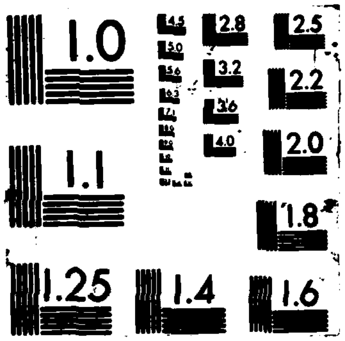
AD-A186 339   ARTIFICIAL INTELLIGENCE/EXPERT SYSTEM COST RESEARCH   1/2
ROADMAP(U) TECOLOTE RESEARCH INC SANTA BARBARA CA
S STEPNEK SEP 86 CR-0145 ESD-TR-87-168 F19628-84-D-0019

UNCLASSIFIED   F/G 12/9   NL


1.0
4.5
5.0
5.6
6.3
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

ESD-TR-87-168 CR-0145

Artificial Intelligence/Expert System Cost Research Roadmap

STEVE STEPANEK

Tecolote Research, Inc.
5266 Hollister Avenue, No. 301
Santa Barbara, California 93111

September 1986

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

Prepared For

ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
DEPUTY COMPTROLLER
HANSCOM AIR FORCE BASE, MASSACHUSETTS 01731

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

ADA186339

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION Unclassified	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION / AVAILABILITY OF REPORT Approved for Public Release; Distribution Unlimited
2b DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) CR-0145	5. MONITORING ORGANIZATION REPORT NUMBER(S) ESD-TR-87-168
6a. NAME OF PERFORMING ORGANIZATION Tecolote Research, Inc. 6b. OFFICE SYMBOL *(If applicable)*	7a. NAME OF MONITORING ORGANIZATION HQ ELECTRONIC SYSTEMS DIVISION (ACCR)
6c. ADDRESS *(City, State, and ZIP Code)* 5266 Hollister Avenue, No. 301 Santa Barbara, California 93111	7b ADDRESS *(City, State, and ZIP Code)* Hanscom AFB Massachusetts, 01731-5000
8a. NAME OF FUNDING / SPONSORING ORGANIZATION Deputy Comptroller 8b OFFICE SYMBOL *(If applicable)* ESD/ACCR	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F19628-84-D-0019
8c. ADDRESS *(City, State, and ZIP Code)* Hanscom AFB Massachusetts, 01731-5000	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO | PROJECT NO | TASK NO | WORK UNIT ACCESSION NO

11 TITLE *(Include Security Classification)*
Artificial Intelligence/Expert Systems Cost Research Roadmap

12 PERSONAL AUTHOR(S)
Steve Stepanek

13a TYPE OF REPORT
Technical

13b TIME COVERED
FROM ________ TO ________

14. DATE OF REPORT *(Year, Month, Day)*
1986 September

15 PAGE COUNT

16 SUPPLEMENTARY NOTATION

17 COSATI CODES

FIELD	GROUP	SUB-GROUP

18 SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Artificial Intelligence, Expert Systems
Cost Research Roadmap

19 ABSTRACT *(Continue on reverse if necessary and identify by block number)*

This roadmap defines the "what, why, when and the priority and cost of the specific tasks to be undertaken" that will enable quality cost estimates for AI/ES projects in the future.

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT
□ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT □ DTIC USERS

21 ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a NAME OF RESPONSIBLE INDIVIDUAL
Joseph Dean, Captain, USAF

22b TELEPHONE *(Include Area Code)*
(617) 377-2679

22c OFFICE SYMBOL
ESD/ACCR

83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

TABLE OF CONTENTS

APPENDIX

LIST OF FIGURES

FIGURE PAGE

LIST OF TABLES

SECTION 1
INTRODUCTION

STUDY OBJECTIVE

Artificial Intelligence (AI) has been around 30 years. System use is increasing at a annual rate of 300% and 1500 systems have been developed and implemented to date. AI system applications include manufacturing, business, professional, and DoD. The computer industry is on the brink of a revolution because AI is transforming computers into intelligent assistants that can perform computational tasks, emulate human thought, recognize patterns, reason solutions to problems, and even explain their conclusions. An integral part of this technology is a new class of computer programs that can incorporate human expertise. These new programs tell the computer what it needs to know and not just what it needs to do.

In the proliferation of AI market studies and claims of technical capabilities, all sorts of messages are being conveyed to the engineering communities. Some claims insist systems are an invaluable technological advance that can replace human beings. Others claim systems solve problems using scientific approaches that are not unique to AI. Uncertainty abounds, yet Business Communications, Stanford, Conn., in its report "Artificial Intelligence: Current and Future Commercialization," maintains AI has a $10 billion potential in the next 10 years. Most of the AI activity has been from Department of Defense Strategic Computing Program, R&D from Fortune 500 Companies, foreign information science projects, Japan's fifth generation computer systems, the European Economic Community ESPRIT Project, Britain's Alvey Directorate, and the Soviet Union's CEMA project. As of now, there are no cost estimating techniques for AI projects.

This Artificial Intelligence/Expert System (AI/ES) Cost Research Roadmap will define the "what, why, when, and the priority and cost of the specific tasks to be undertaken" that will enable quality cost estimates for AI/ES projects in the future. The purpose of this study is to identify AI cost estimating problems and needs, "formulate research tasks to meet ESD estimating needs (and) structure the tasks into a coherent roadmap..." These tasks are candidate for Electronic Systems Division funding as future research topics.

A the contractor

STUDY APPROACH

Tecolote Research, Inc. was tasked to obtain an overall understanding of Artificial Intelligence/Expert Systems and formulate research projects for developing AI/ES cost models. The five-steps involved are the following:

1. Identify and understand AI/ES technical requirements. Tecolote will compile a list of AI vendors included as Appendix C and compile a list of experts from government and industrial organizations included as Appendix B. An AI/ES data search will require literature and text books to be reviewed and compiled as a Bibliography included as Appendix A.

2. Interview experts. As many as possible of the experts listed in Appendix B will be interviewed. These managers and engineers will include military and non-military personnel. Each was selected to be interviewed based on their extent and variety of background and diversity of AI knowledge and interest. This interview sample approach produced a one of a kind, ground up source of information not available by any other means. Topics discussed included AI state of the art (SOA), previous projects the interviewee developed, lessons learned from those projects, current AI strengths and weakness, technology breakthroughs and research studies required to advance SOA. Each interview and telephone conversation is reported and included as Appendix D.

3. Identify Technology Areas. Items identified during the interview or data search process will be organized into technology areas. These areas will include technical impact on system performance and cost impact on system efficiency if fully implemented.

4. Identify Research Areas. Since AI/ES is an emerging technology, continuing and on-going research will be needed to identify and develop quality cost estimating tools. These research areas will identify when the research data may be available and the significance and impact that that data will have on understanding AI system development cost.

5. Provide Specific Recommendations. Organize these technology and research areas into a prioritized list of study area candidates as future ESD tasks including an estimated cost to achieve each.

STUDY CONDUCT

After reading much of the available literature, it became evident that the definition of Artificial Intelligence is as diverse as the methods to develop "it" or the techniques to manage "it." The AI cost estimating problem seemed to pale in the face of how to develop and manage the AI effort. Many of the first interviews concluded with "Good luck, you can't cost estimate AI projects anyway. If you learn anything about how to develop an AI system using an engineering problem solving approach, let us know. AI is so different that previous cost models don't apply and conventional software development techniques don't work."

What also did become evident is the importance to develop a cost research roadmap that does not preclude previous work performed that estimates conventional (non-AI) software projects. Some conventional cost estimating models do not apply to AI projects but most models do apply if modified. As a result, research and technology areas were identified that would provide data and cost parameters to conventional models that would then be able to estimate AI projects. The rule of thumb was not to re-create the wheel but rather modify it as needed.

Subsequent interviews and reading focused on the approach to modify existing software cost models to estimate AI projects. Nine Technology areas that will impact AI system performance, efficiency, and cost were areas that will impact AI system performance, efficiency, and cost were listed. Also listed were sixteen research areas that when performed and data analyzed, will provide an understanding of AI cost and schedule estimating. These ideas are discussed in Sections 4 and 5 and each area includes the following:

Definition - a short, precise definition of the area.

Significance - a qualitative statement about the significance of the research area.

Discussion - a free flowing discussion of the area.

Related Areas - Identification of how this area relates to other areas listed in this report.

Doability - an assessment of the difficulty to achieve success in the research effort.

Priority - a rating of the importance that the cost estimator understand the impact of this area and is graded as extreme, high, medium, or low.

Recommendation - a recommendation as to whether or when the research topic should be pursued, based on priority.

Research Steps - an outline of steps for future research projects.

REPORT ORGANIZATION

The remainder of this report contains six sections. Section 2 contains AI definitions, the state of the art techniques, and current applications. Section 3 contains a list of future AI capabilities. Section 4 contains AI technology areas that, if and when implemented, will impact AI cost estimating. Section 5 contains study areas to identify and gather data needed to develop AI cost estimating tools. Section 6 contains final recommendations concerning future tasks.

Section 4 describes AI technology methods and techniques that current cost estimating tools do not evaluate or utilize. The cost analyst has to make sure estimating tools will exist that can evaluate system performance, efficiency, and cost as these technologies become understood or available. The nine technology areas discussed in Section 4 are as follows:

T1. VHSIC. This section identifies the cost impact that this technology program will (1) solve AI processing speed and throughput requirements and (2) solve the miniaturization requirement for complex hardware architectures.

T2. Problem Solving Techniques. Different AI problems require different methods to access data and process programming statements. This section identifies the cost trade-off of these various methods.

T3. Knowledge Representation. AI .ystems utilize knowledge data systems that must be organized to best accommodate the processing technique used for each problem. This section described the cost impacts of different methods to represent data.

T4. AI Languages. There are new high order languages (HOL) that, if developed, will enhance the efficiency of developing AI products and therefore impact the resulting development cost.

T5. Commercial Tools. The availability of commercial tools impacts the ease to develop and maintain AI systems. System cost is directly affected by the use of these tools.

T6. System Specifications. AI development techniques preclude the use of formal specification for describing system requirements. Without these specifications, cost estimating and systems verification are meaningless.

T7. Development Methodology. AI program development methodology requires a new phase definition in order to perform cost accounting. Quality cost estimating requires this cost data.

T8. Schedule/Cost Estimating. This section describes various AI development cost drivers that must be analyzed for software CERs.

T9. Technology Roadmap. This section describes new AI technologies that the cost estimator must research to prepare for estimating the cost impact.

Section 5 identifies and examines parameters that are AI development cost drivers that need to be used as inputs to cost models when estimating AI projects. This section recommends sixteen research projects to gather and analyze data and develop CERs. These research projects are as follows:

R1. Expanding Present Cost Models. This section identifies factors considered as parameters to present cost models that need to be adjusted and extrapolated when estimating AI projects.

R2. AI Cost Data Base. This section defines data sets required by the cost estimator to model AI project costs.

R3. Schedule Estimating. This section identifies data required by the cost estimator to predict the time that will be required to develop AI projects.

R4. Quality Estimating. This section describes the need for QA methods that apply to AI projects to ensure goals are achieved, on schedule, in cost.

R5. Integration Cost. This section recommends study areas needed to determine the cost to integrate AI software.

R6. Maintenance Cost. This section recommends research needed to predict the cost to maintain AI software as a relationship to the total life cycle cost of AI software.

R7. Security Cost. This is an SDI related subject where research is needed to predict the cost impact of security on AI software projects.

R8. Computer Trade-off. This research area identifies the need to analyze computer costs and processing capabilities necessary for large, complex AI requirements.

R9. Level of Effort Cost. AI development methods use phase distribution of labor that is not understood and therefore is not yet predictable by the cost estimator.

R10. Knowledge Capture Cost. This section describes the research necessary to predict the cost to acquire, understand, and format the knowledge base that will be processed during AI system execution.

R11. Memory Management Cost. This section describes study areas to anticipate AI system overhead cost that support dynamic memory architectures necessary during AI system processing.

R12. Interface Cost. This section describes data required to predict the cost of the expert system interfaces depending on number of interfaces, its complexity, function, and capability.

R13. AI Language Proficiencies. This section identifies research necessary to predict which AI language to use to most efficiently solve any class of AI application.

R14. Tools. This section recommends research that will determine which tool to use as a development aid when producing AI products.

R15. Personnel Considerations. AI programmers are a new and rare breed of computer scientist. The cost estimator will need to predict the additional costs to attract, train and retain these individuals.

R16. Program Management. This section identifies study areas that when understood will help AI managers perform make/buy decisions, estimate development schedules and costs, and be able to measure system progress.

SUMMARY OF RECOMMENDATIONS

Each Technology Area in Section 4 and Research Area in Section 5 was evaluated for its priority and doability as shown in Table 1.1. Priority is the rating of the importance that the cost estimator understand this cost impact and is graded as extreme, high, medium, or low. Doability is the assessment to achieve success in understanding each research effort.

Table 1.1
SUMMARY OF PRIORITY/DOABILITY

DISCUSSION AREA		PRIORITY	DOABILITY
T1	VHSIC	High	Possible
T2	Problem Solving Techniques	Extreme	Very Achievable
T3	Knowledge Representation	Extreme	Very Achievable
T4	AI Languages	Low	Possible
T5	Commercial Tools	Extreme	Very Achievable
T6	System Specifications	Extreme	Possible
T7	Development Methodology	Extreme	Very Achievable
T8	Schedule/Cost Estimating	High	Achievable
T9	Technology Roadmap	Extreme	Very Achievable
R1	Expanding Present Cost Models	Extreme	Achievable
R2	AI Cost Data Base	Extreme	Achievable
R3	Schedule Estimating	High	Possible
R4	Quality Estimating	High	Low
R5	Integration Cost	Low	Achievable
R6	Maintenance Cost	Low	Achievable
R7	Security Cost	Medium	Possible
R8	Computer Trade-off	Low	Very Achievable
R9	Level of Effort Cost	High	Achievable
R10	Knowledge Capture Cost	High	Achievable
R11	Memory Management Cost	Low	Low
R12	Interface Cost	Medium	Possible
R13	AI Language Proficiencies	High	Very Achievable
R14	Tools	High	Very Achievable
R15	Personnel Considerations	Medium	Achievable
R16	Program Management	High	Possible

Section 6 describes in detail the various factors considered in determining when to initiate each study and how many man-months are estimated to adequately perform the research. Table 1.2 summarizes, by year the recommended effort that a Sr. Cost Analyst would require to study each area and develop AI software CERs.

Table 1.2
RESEARCH/STUDY LOE ESTIMATES

(Man months of Senior Cost Research Labor)

DISCUSSION						
AREA		1987	1988	1989	1990	1991
T1	VHSIC		3	3	1	1
T2	Problem Solving Techniques	3	6	3	1	1
T3	Knowledge Representation	3	6	6	3	3
T4	AI Languages				3	2
T5	Commercial Tools	6	3	3	2	2
T6	System Specifications	6	6	3	3	3
T7	Development Methodology	6	6	3	3	3
T8	Schedule/Cost Estimating		3	3	3	3
T9	Technology Roadmap	3	3	3	2	1
R1	Expanding Present Cost Models	6	12	6	6	6
R2	AI Cost Data Base	24	24	12	12	12
R3	Schedule Estimating		3	3	2	2
R4	Quality Estimating		3	6	3	3
R5	Integration Cost				3	2
R6	Maintenance Cost			3	2	1
R7	Security Cost			3		
R8	Computer Trade-off				3	
R9	Level of Effort Cost		3	2		
R10	Knowledge Capture Cost		6	3		
R11	Memory Management Cost				1	
R12	Interface Cost			2		
R13	AI Language Proficiencies		3	3		
R14	Tools		3	2	1	1
R15	Personnel Considerations			2		
R16	Program Management		3	2		

SECTION 2
OVERVIEW OF ARTIFICIAL INTELLIGENCE

The purpose of this AI Cost Research Roadmap is to define the what, why, when, and the priority and cost of the specific tasks to be undertaken that will enable quality cost estimates for AI projects in the future. The purpose of Section 2 is to explain why knowing about AI is important, to outline the field, and to give some examples of progress made to date. In order, this section includes the following: the evolution of AI, AI systems implemented by 1986, AI terminology and definitions, how to develop AI programs, and current AI problem areas. This section will provide an understanding of the AI state of the art (SOA).

EVOLUTION OF AI

As World War II ended, the British and American scientists were developing what we now call a computer that would be an electronic machine guided by a stored program of directions that would perform complex arithmetic computations. As shown in Figure 1, psychologists were simultaneously thinking that computers could be developed that would mimic human behavior by manipulating non-numerical symbols.

By 1960, certain successes were achieved. A program for solving geometric analogy problems like those on intelligence tests was developed. Another program that did symbolic integration was the forerunner to today's MACSYMA and other mathematics manipulation system. These two examples, integration and analogy introduced the ideas that have become expert systems.

During 1965 to 1970, AI development was slowed because the tremendous enthusiasm of earlier was dampened as expected results had not been produced. This was a time when people searched for general problem solvers, that is, a mechanism that when placed in a computer would process data and appear intelligent.

During 1970 to 1975, successful projects were achieved. LISP, a programming language designed for AI applications and still very popular today, was implemented. Robotics concepts were demonstrated and MYCIN, a program to aid physicians diagnose patients and prescribe medicines, was developed.

During 1975 and 1980, knowledge based system technology was understood and AI was acknowledged as a specialized subset of computer sciences. Systems such as HEARSAY II, MACSYMA, EMYCIN, DENDRAL were developed and a second AI language named PROLOG was implemented. AI was recognized as the generic methodology composed of five problem solving application areas: expert systems, vision processing, robotics, natural language processing, and automatic programming.

From 1980 to the present, commercial AI companies appeared and flourished and many AI applications have been developed. Examples will be

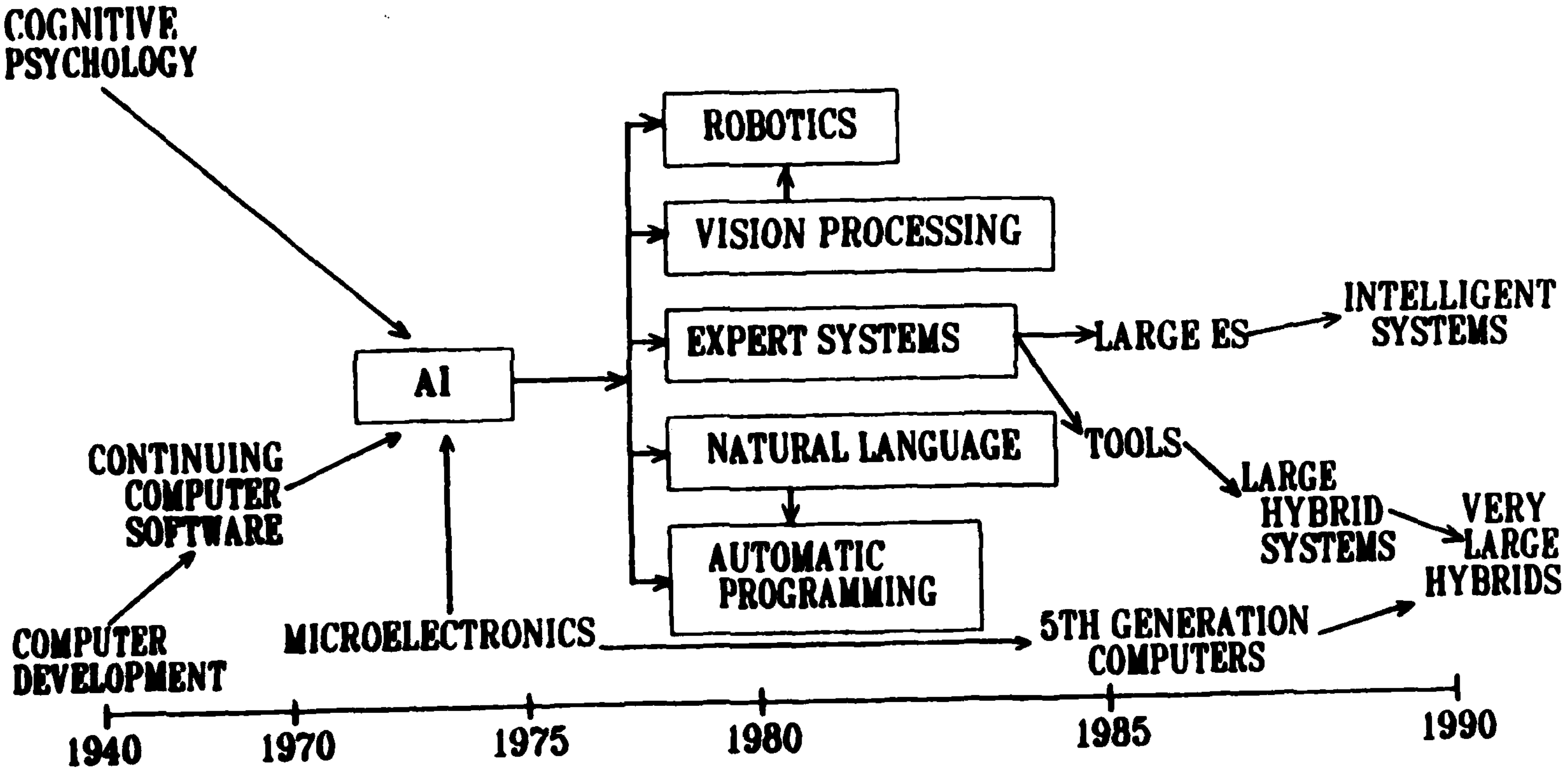


FIGURE 1
THE AI EVOLUTION

Source: Expert Systems by Paul Harmon and David King, John Wiley & Sons, Inc.

discussed in the next paragraph as AI systems implemented by 1986. Figure 1 projects AI technology advances beyond the present. Section 3 will discuss anticipated AI applications and technology breakthroughs.

AI SYSTEMS IMPLEMENTED BY 1986

Artificial Intelligence is the part of computer sciences concerned with designing computer systems that exhibit the characteristics we associate with intelligent human behavior. AI, at present contains five applications areas: expert systems (where almost all of the AI effort is focused), vision processing, robotics, natural language processing, and automatic programming. Each area and examples of programs implemented are discussed in the following paragraphs.

1. EXPERT SYSTEMS (ES) is the branch of AI that performs tasks normally performed by human experts. Human experts and expert systems specialize in narrow problem solving tasks. Both can solve problems relatively easily, explain what they do, know when they are stumped, communicate with other experts, learn from experience, change their points of view, transfer knowledge, reason using tools, rules of thumb called heuristics, and math models. Successful examples are:

   o MYCIN was developed by Edward Shortliffe at Stanford in the mid 1970s. It provides consulting advice about infections to provide physicians advice comparable to what they would receive from a physician specializing in bacteria and meningitis. This system is one of the best known and most widely used expert systems.

   o XCON was developed by Carnegie Mellon University for Digital Equipment Corporation in 1978. It has been enhanced continually to now be able to validate any hardware configuration for any DEC computer. This system has been widely studied as a case history of ES development, test, and maintenance techniques.

   o MACSYMA was developed by Joel Moses at MIT. This ES was the first attempt to integrate math manipulations and symbolic processing by helping students perform complicated math applications.

   o DENDRAL was developed at Stanford in the early 1970s to determine organic structures using data from mass spectrometers and nuclear magnetic resonance machines.

   o EMYCIN was developed at Stanford as the first attempt to make an existing ES (MYCIN) a general purpose ES. EMYCIN is the non-specific part of MYCIN consisting of what is left when the rules are removed. EMYCIN becomes a new ES by adding rules for a different problem domain.

   o Many applications have been developed for business/financial/legal systems that provide investment risk assessment, portfolio management, tax and financial planning, office management, and data center management.

- o MATCALS developed by Marine Corps can automatically control and land an F-18.

2. VISION PROCESSING is the branch of AI that can perceive the location, form, size, shape, and color of objects. They can process and interpret images from remotes sensors or perform as the eyes for robots. Very little effort is currently being spent to advance vision processing SOA. Successful examples are:

   - o INDUCE developed by Ryszard S. Michalski can distinguish between different types of railroad cars.

   - o PROSPECTOR can draw contour geological survey maps from aerial images.

   - o Some applications have been developed to coordinate robot hand and eye movement, can see faulty products manufactured on an assembly line, can spot and remove diseased soy beans as they pass on a conveyer belt.

3. ROBOTICS is the branch of AI that enables computers to see using vision processing and then manipulate objects. Robotics can use heuristics (logic rules of thumb) and function in a highly flexible manner while interacting with a constantly changing environment. Successful examples are:

   - o An autonomous land vehicle built by Martin Marietta for Defense Advanced Research Projects Agency can navigate a road and avoid obstacle at 10 m.p.h.

   - o Robots can fight dangerous fire such as at Three Mile Island.

   - o Robots can perform dirty, dangerous, and boring manufacturing tasks.

   - o Robots developed by Rodney A. Brooks perform spatial reasoning by moving objects among and through a cluttered environment.

4. NATURAL LANGUAGE PROCESSING is the branch of AI that can process and interpret natural languages such as English, Spanish, or French and input or output in both text and sound. Successful examples are:

   - o INTELLECT developed by Larry R. Harris can understand questions expressed in English like "How did forecast sales compare to actuals last month?" and then print a report.

   - o HEARSAY II developed in the early 1970s can analyze verbal questions and commands, determine what was said, and update the appropriate fields in a data base or management information system.

   - o LIFER developed by Gary G. Henrix is capable of answering questions about ships like "What is the length of the CVA-67?"

- o Systems can read documents and either store them as internal files or translate them to another natural language.
- o Systems developed by Kurtzweil can read and then speak documents to blind people eliminating braille writing.

5. AUTOMATIC PROGRAMMING is the branch of AI that can automatically generate computer software. Successful examples are:

- o USE.IT developed by High Order Software, Inc. can help the user develop a logically sound flowchart of a system and then create the correct FORTRAN and Pascal programs.
- o Application programs can automatically detect syntax errors as lines of code are input by the programmer.

AI DEFINITIONS

AI is a new and evolving field within computer sciences and it utilizes a vocabulary that is also new and evolving. Many terms in AI have a variety of diverse definitions where each definition has a loyal and typically emotional following of backers. The definition used below is not the only one available but will be the definition used for the purpose of this report.

Artificial Intelligence. The part of computer sciences concerned with designing computer systems that exhibit the characteristics we associate with intelligent human behavior. AI at present contains five application areas: expert systems (where almost all of the AI effort is focused), vision processing, robotics, natural language processing, and automatic programming.

Automatic Programming. The branch of AI that can automatically generate computer software.

Backward Chaining. That problem-solving technique characterized by working backward from hypothesized conclusions toward known facts.

Blackboard Agenda. An ES design in which several independent knowledge bases can each examine a common working memory called a blackboard.

Breadth-first Search. In a hierarchy of rules or objects, breadth-first search refers to a strategy in which all of the rules or objects on the same level of the hierarchy are examined before any of the rules or objects on the next lower level are checked.

C. Popular programming language, especially for systems programming.

Common LISP. A dialect of LISP that is intended to serve as a standard version of LISP that will run on a number of different machines.

Context-Parameter-Value Triplets. (Object-Attribute-Value Triplets.) One method of representing factual knowledge; it is the method used in EMYCIN. A context is an actual or conceptual entity in the domain of the consultant (e.g., a patient, an aircraft, or an oil well). Parameters are properties associated with each context (e.g., age and sex of a patient or location and depth of an oil well.) Each parameter (or attribute) can take on values: the parameter, age, could take the value "13 years."

Control (of a knowledge system). The method used by the inference engine to regulate the order in which reasoning occurs. Backward chaining, forward chaining, and blackboard agendas are all examples of control methods.

Depth-first Search. In a hierarchy of rules or objects, depth-first search refers to a strategy in which one rule or object on the highest level is examined and then the rules or objects immediately below that one are examined. Proceeding in this manner, the system will search down a single branch of the hierarchy tree until it ends. This contrasts with breadth-first search.

Domain. A topical area or region of knowledge. Medicine, engineering, and management science are very broad domains. Existing knowledge systems only provide competent advice within very narrowly defined domains.

Expert Systems. The branch of AI that performs tasks normally performed by human experts. Human experts and expert systems specialize in narrow problem solving tasks. Both can solve problems relatively easily, explain what they do, know when they are stumped, communicate with other experts, learn from experience, change their points of view, transfer knowledge, reason using tools, rules of thumb called heuristics, and math models.

Explanation. Broadly, this refers to information that is presented to justify a particular course of reasoning or action. In knowledge systems this typically refers to a number of techniques that help a user understand what a system is doing.

Fifth-Generation Computers. The next generation of computing machines. It is assumed that they will be larger and faster and will incorporate fundamentally new designs. Parallel processing, the ability of a computer to process several different programs simultaneously, is expected to result in a massive increment in computational power.

Forward Chaining. Problem-solving technique characterized by working forward from known facts toward conclusions.

Frame. (Object or Unit.) A knowledge representation scheme that associates an object with a collection of features (e.g., facts, rules, defaults, and active values). Each feature is stored in a slot. A frame is similar to a property list, schema, or record, as these terms are used on conventional programming.

Heuristic. A rule-of-thumb or other device or simplification that reduces or limits search in large problem spaces. Unlike algorithms, heuristics do not guarantee correct solutions.

Heuristic Rules. Rules written to capture the heuristics an expert uses to solve a problem. The expert's original heuristics may not have taken the form of if-then rules, and one of the problems involved in building a knowledge system is converting an expert's heuristic knowledge into rules. The power of a knowledge system reflects the heuristic rules in the knowledge base.

Inference. The process by which new facts are derived from known facts. A rule (e.g., If the sky is black, then the time is night), combined with a rule of inference (e.g., modus ponens) and a known fact (e.g., The sky is black) results in a new fact (e.g., The time is night).

Inference Engine. That portion of a knowledge system that contains the inference and control strategies. More broadly, the inference engine also includes various knowledge acquisition, explanation, and user interface subsystems. Inference engines are characterized by the inference and control strategies they use.

Interface. The link between a computer program and the outside world. A single program may have several interfaces. Knowledge systems typically have interfaces for development (the knowledge acquisition interface) and for users (the user interface). In addition, some systems have interfaces that pass information to and from other programs, data bases, display devices, or sensors.

Knowledge Acquisition. The process of locating, collecting, and refining knowledge. This may require interviews with experts, research in a library, or introspection. The person undertaking the knowledge acquisition must convert the acquired knowledge into a form that can be used by a computer program.

Knowledge Base. The portion of a knowledge system that consists of the facts and heuristics about a domain.

Knowledge Engineer. (Knowledge Engineering.) An individual whose specialty is assessing problems, acquiring knowledge, and building knowledge systems. Ordinarily this implies training in cognitive science, computer science, and artificial intelligence. It also suggests experience in the actual development of one or more expert systems.

Knowledge Representation. The method used to encode and store facts and relationships in a knowledge base. Semantic networks, object-attribute-value triplets, production rules, frames, and logical expressions are all ways to represent knowledge.

Knowledge System. A computer program that uses knowledge and inference procedures to solve difficult problems. The knowledge necessary to perform at such a level, plus the inference procedures used, can be thought of as a model of the expertise of skilled practitioners. In contrast to expert systems, knowledge systems are often designed to solve small, difficult problems rather than large problems requiring true human expertise.

LISP. A popular language for programming AI.

Natural Language Processing. The branch of AI that can process and interpret natural languages such as English, Spanish, or French and input or output in both text and sound.

Operating System. The computer software system that does the "housekeeping" and communication chores for the more specialized systems.

Parallel Processing. A proposed architecture for computer machinery that would allow a computer to run several programs simultaneously. It would mean that a computer would have several central processors simultaneously processing information.

PROLOG. A symbolic AI programming language.

Prototype. In expert systems development, a prototype is an initial version of an expert system. It is usually a system with from 25 to 200 rules that is developed to test effectiveness of the overall knowledge representation and inference strategies being employed to solve a particular problem.

Pruning. In expert systems, this refers to the process whereby one or more branches of a decision tree are "cut off" or ignored. In effect, when an expert system consultation is underway, heuristic rules reduce the search space by determining that certain branches (or subsets of rules) can be ignored.

Robotics. The branch of AI that enables computers to see using vision processing and then manipulate objects. Robotics can use heuristics (logic rules of thumb) and function in a highly flexible manner while interacting with a constantly changing environment.

Rule. A conditional statement of two parts. The first part, comprised of one or more if clauses, establishes conditions that must apply if a second part, comprised of one or more then clauses, is to be acted upon.

Rule-based Program. A computer program that represents knowledge by means of rules.

Symbol. An arbitrary sign used to represent objects, concepts, operations, relationships, or qualities.

Symbolic versus Numeric Programming. A contrast between the two primary uses of computers. Data reduction, data-base management, and work processing are examples of conventional or numerical programming. Knowledge systems depend on symbolic programming to manipulate strings of symbols with logical rather than numerical operators.

Vision Processing. The branch of AI that can perceive the location, form, size, shape, and color of objects. They can process and interpret images from remotes sensors or perform as the eyes for robots.

CONVENTIONAL PROGRAMMING VS. EXPERT SYSTEM PROGRAMMING

The skeptics feel that expert systems use sound scientific principles and approaches to solve their problems but they are not so magic that conventional programs could not also provide the same solution. This is true with qualifications. A conventional program can produce the same results today that an expert system can but expert systems have a theoretical scope of problems that can be solved that is much larger and more complex than that of any conventional system. Not all the ES technology and methodology is yet understood or perfected, but if and when that does occur, solutions will be available to problems that experts cannot even imagine. High order languages replaced assembly languages and allowed computers to solve more complex problems. These new systems were not only more powerful and proficient but were cheaper to build, faster to implement, easier to test, and more simple to enhance. Now, AI is replacing conventional programs with systems that can solve even more complex problems and do it using tools that allow more efficient programming techniques, faster program development and test turnaround, and all for less money. In practice therefore, a conventional program can probably solve any problem an expert system can but not as efficiently. In theory, the scope of problem solved by an expert system is immense and the effort to achieve that success is minimal. For that reason, expert system research will continue. Because of the successes already demonstrated, expert system feasibility can be proven.

Conventional Programming (also called Third Generation Programming or Traditional Programming) consists of interconnected procedures and sequential processing of computer statements. Systems are developed using a waterfall methodology consisting of a sequential series of development phases as shown in Figure 2. Each phase has subgoals and is considered complete by a verification and validation activity. Each phase then is the input for a successive phase.

Conventional programming techniques have been used to create the large data processing systems we associate with computers. These systems collect large volumes of data stored in a data base and process this data using numerical computations and arithmetic algorithms. These algorithms are a step-by-step procedure that guarantee the right conclusion will be reached when processing the correct data. For example, each evening all of the data regarding all of the changes in every account at your local bank are fed into a computer. Then a very complex algorithm works through the data, making additions and subtractions, calculating the proper fee, and finally arriving at the bank's overall balance for the day. The numbers differ each day, but they are always processed in the same way, and they always result in a predetermined conclusion - the bank's overall balance.

Expert systems are programmed by knowledge engineers using very different techniques. The goal of the ES is to mimic human experts as opposed to processing predetermined, sequential program operations that produce predictable results. The knowledge engineer interacts with the human expert to help describe the knowledge as data and then programs inference rules that process the data. The knowledge engineer combines

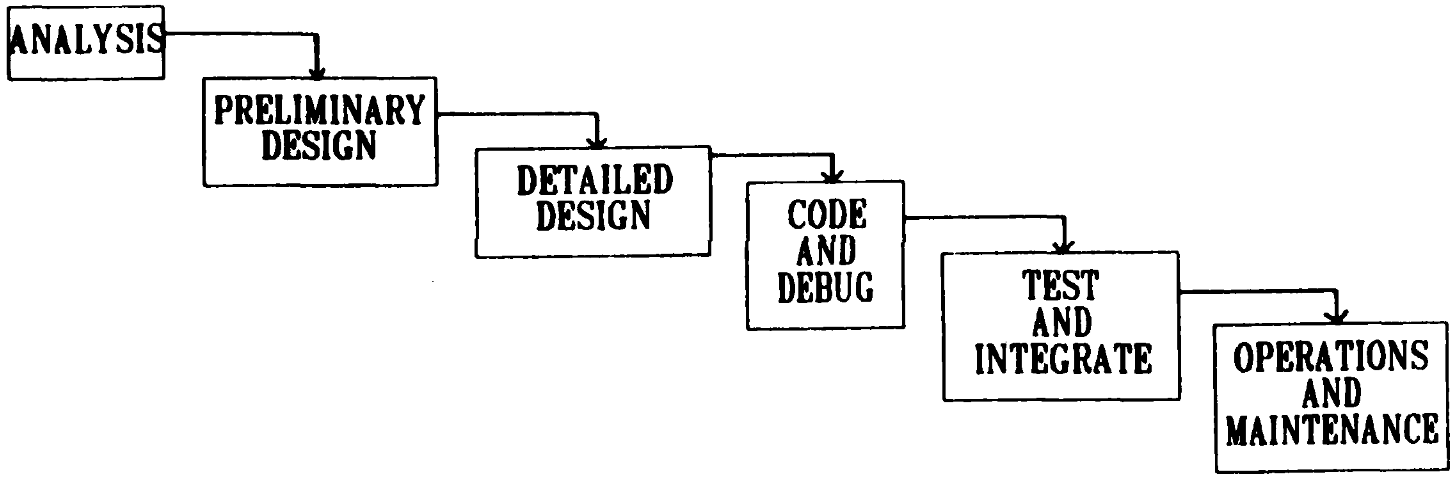


FIGURE 2
WATERFALL METHOD OF CONVENTIONAL SOFTWARE DEVELOPMENT

cognitive psychology with symbolic programming techniques to develop the expert system.

Knowledge engineers (KE) use a interactive technique to develop the ES by meeting often with the expert. During the first meeting, they seek a first cut at the problem by implementing a prototype system with only a few rules and a few facts. This system is tested and the expert is asked more questions. The second system version is implemented using more rules and more facts. This system is tested and the expert is again asked more questions. This process continues as a series of approximations where each is more complex and refined than the previous until the system produces the desired results. Conventional programs approach their tasks in ways that contrast to expert systems. These contrasts are as follows:

CONVENTIONAL PROGRAMS	EXPERT SYSTEMS
1. Data schemes are flat and do not enable a hierarchy of concepts based on different abstraction levels.	1. Frames contain data descriptions and instances that represent concepts and relationships between abstraction levels. Both objects and complex entities can be described that describe processes to be performed or plans that must be followed.
2. Programs define what is to be accomplished and a step-by-step procedure to achieve the result.	2. ES defines the goal to be achieved and the program explores logic alternatives to figure out how to achieve it.
3. Programs cannot improve themselves.	3. ES can solve more diverse problems each time it executes by remembering rules and logic relationships from previous problems solved.
4. Not very interactive. If a non-programmer stopped the run and examined the code, he could not understand very much.	4. Very interactive. The user can halt the system and ask questions that can provide recommended conclusions to be attempted next.
5. Processing by predetermined algorithms.	5. Processing with heuristics that modify the approach and reduce the limit search depending on the conclusion of previous rules.
6. Data in data base is readable only by programmers.	6. Data is easy to read and easy to modify.
7. Most of the programmer's time is spent developing the procedural structure of the system so the program can deal with the right problems and issues at the right time.	7. Most of the programmer's time is spent developing the facts, concepts, and rules to be used rather then organizing and proceduralizing it.
8. Systems are designed from predefined specifications.	8. Systems can be developed whose specifications can not be fully determined.
9. Systems are designed to implement operational requirements through formal feedback.	9. System requirements are dynamic and can be based on incoming data or commands.
10. System maintenance is difficult.	10. Systems based on thousands of alternatives that are easy to maintain.

PROGRAMMING EXPERT SYSTEMS

Expert systems are rule-based programs that consist of a large number of IF-THEN rules where each is an independent piece of code. The IF part tests the rules for a condition, and if the condition is met, the THEN actions are activated. Unlike statements in traditional programs, a rule's actions don't necessarily invoke another rule but may instead modify the symbolic data base which contains all the application's current statements.

An expert system contains five parts as shown in Figure 3: input, output, inference engine, parser, and knowledge base and they are defined as follows:

1. Input is from the user, knowledge engineer, device or another system. The user can provide commands, ask questions about what data was used or which rules were used to arrive at a particular conclusion, or answer questions the ES may ask during an interactive program run. The knowledge engineer can provide data, format the structure of the data, program rules and define the sequence that rules can be considered.

2. Output is to the user, knowledge engineer, device or another system. The user can receive problem solution or recommendations, program explanations, or interactive responses. The knowledge engineer may query the system during test runs.

3. Inference engine contains the inference (rules) and control (order that reasoning occurs) strategies so facts are processed and results produced.

4. Knowledge Base is developed by the KE using the experts' knowledge and it contains rules and facts that solve problems.

5. Parser converts rules to commands that are understandable by the inference engine (this is the analogy to a compiler).

The ES players are:

1. Human expert.
2. Knowledge Engineer who is a computer scientist that uses the information of the expert to write the rules and facts in the knowledge base.
3. The user who has the problem to be solved.

Expert Systems are developed by interviewing the expert and the knowledge engineer writing the rules and the facts. Experts solve problems by capturing a large number of cases and applying them to data. Rules are an effective way to encode this expertise and rules can be rapidly constructed and rapidly tested via direct user feedback. This freedom from procedural design produces rapid logic construction through user feedback and is defined as rapid prototyping.

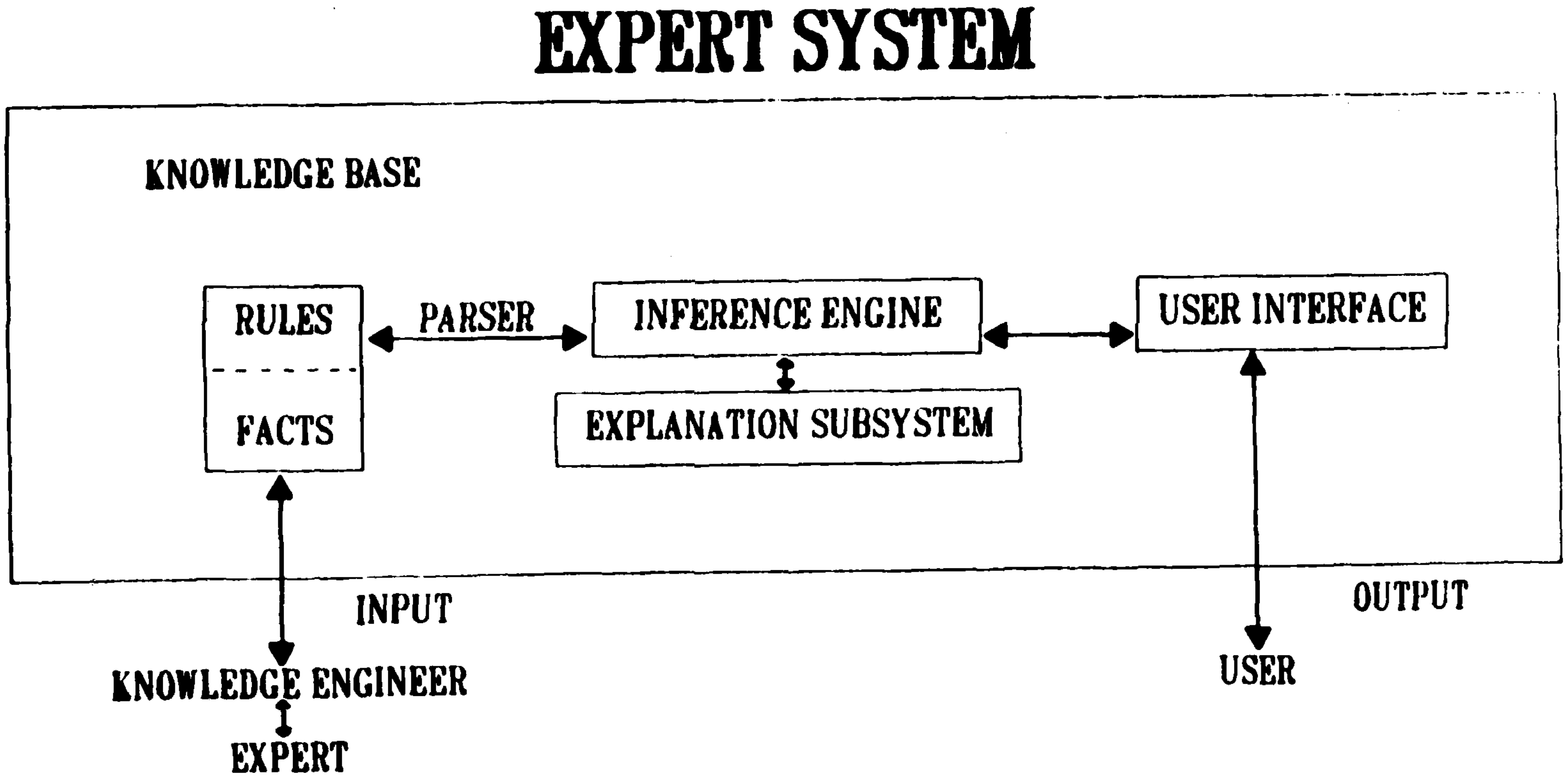


FIGURE 3
EXPERT SYSTEM WITH INTERFACES

Rapid prototyping allows "build a little, test a little" development through constant feedback. Changes are easily implemented, complete new sets of rules are added so that as the system is developed the evolving application is always operational. Changes are easy to make because program control is based on rules and not strict, third generation procedural organization. Changing the specifications, adding or modify capabilities, or improving process effectiveness by the developer's enhanced understanding of the problem is possible by rapid prototyping. Acquiring a complete set of rules to emulate the problem solving of a human expert is impossible but rapid prototyping ensures that the developed application has the maximum user effectiveness.

CURRENT EXPERT SYSTEM PROBLEMS

The following problem areas were identified by interviewing developers and managers who have practical experience with expert systems:

1. The need for an ES validation tool. There is no way to absolutely confirm or predict results from an expert system.

2. The need for development standards or programming conventions for AI systems. If DoD decides to utilize ES techniques, they will need a corollary to MIL-STD-2167.

3. The need for a knowledge base maintenance tool that performs rapid updates during prototyping.

4. The need for a debug tool that supplies appropriate why and how explanation of ES behavior and conclusions.

5. The need to reconcile conflicting conclusions during subsequent execution runs.

6. The need for a method of representing time dependent or time backward data.

7. The need to qualify data with a plausibility and possibility factor.

8. The need for a commercial tool that assists in knowledge acquisition and provides tight verification and validation of the reasoning process.

9. The need for coding and planning conventions to reduce the number of development iterations. This can eliminate the "develop five, throw away four system" prototyping approach.

10. The need for cost data and cost estimating models for software projects larger then 900K SLOC.

11. The need for more ES programmers and more experts available to them.

12. The need to adequately test ES realtime systems under realistic, actual use conditions.

13. The need for a tool that develops and maintains a knowledge base.

14. The need to formally specify AI requirements.

15. The need for ES interactive debug aids.

16. The need for special purpose, parallel, and analog computers.

17. The need to determine software quality of ES systems.

18. The need for a tool to estimate the phase distribution of labor for ES systems.

19. The need to represent knowledge in a natural way in order to maintain it and read it by the user, knowledge engineer, and expert.

20. The need to scope ES development and implementation schedules and cost.

SECTION 3
FUTURE ARTIFICIAL INTELLIGENCE APPLICATIONS

Section 2 presented the evolution of AI to the year 1986 and explained why knowing about AI is important, outlined the field, gave examples of successful systems implemented, and identified current problem areas that need to be solved to push technology. This section will present the anticipated AI applications shown in Figure 1. These applications should evolve by 1995. By knowing where AI will go, the reader will gain a pragmatic understanding and appreciation of the Technology and Research areas described in the following sections. This section will provide an insight to the future of AI by listing what products will probably be available within the next ten to fifteen years.

Figure 1 shows that the major research efforts and resulting advances in capabilities will occur in the expert system part of AI. Work in other parts of AI will continue but significant advance in their state of the art are not expected. Natural Language Processing capabilities will be stopped by the technical barrier to understand those expressions we humans take for granted such as humor, cynicism, worldly knowledge, and intuitive association of subject matter (relating to previous discussions and conversation). Automatic Programming capabilities will not advance much further then what is presently available.

Robots will see utilizing telerobotics where they are manipulated by a remote operator that sees using television cameras for eyes. Vision processing will not progress much until the new fifth generation computers are available that can process the incredible volume of data necessary to distinguish holes, edges, and shadows in realtime.

Fifth generation machines are expected during 1990 that will be larger, faster, and incorporate fundamentally new design. These computers will advance the AI SOA because they will be able to perform parallel processing (the ability to process many different programs simultaneously) which is expected to result in a massive increase in computational speed and power. Since expert systems will tend to be very large and involve large amounts of processing, it is assumed they will not reach maturity until these computers become available.

Expert systems beyond 1990, are expected to deliver computer solved answers to problems that most experts cannot now imagine. Truly intelligent systems will become available and manage command and control functions necessary to coordinate the battlefield. Expert systems that can teach students directly or assist trainers is an application expected to deliver a high payoff because students will learn more and faster and it will free teachers to work one-on-one with individual students. The following lists six AI areas that will develop capabilities over the next 10-15 years to include:

NATURAL LANGUAGE PROCESSING

1. Systems will make progress in the area of text interpretation, speech understanding, and language translation. The major barrier to understanding natural language is that large amounts of common sense, intuition, and worldly knowledge are required. Highly structured and stylized languages such as those used by air traffic control will be the first to be implemented.

2. Systems will be able to automatically record verbal communications and automatically supply supplementary data as needed for clarification and understanding. This AI system will need to modify the level of explanation it supplies. As human beings, we know about things at different levels of details. Depending on how much the other person knows, our explanations are modified.

3. Systems will be able to solve problems by analogy from rules created by previous tasks. The problem of inferencing (what was meant or implied by what was said) will be a barrier. The initial problems will need to be straight forward and highly structured in presentation.

4. Voice printing and voice recognition will provide better security.

ROBOTICS

1. Systems will support space missions as unmanned, autonomous vehicles. They could be given a list of missions and be allowed to choose between them depending on the situation.

2. Systems will be surveillance sensors that can distinguish objects of interest in a cluttered background in all weather. Sensors will not be limited to line of sight.

3. Robots will perform battlefield ammunition supply and other dangerous, "smart" tasks like removing mines or bombs.

4. Sophisticated vision systems will substitute the human brain, hand, eye and arm in the assembly plant. These systems will inspect, clean, assemble, and provide process control for productions tasks such as welding, foundry, bolting, painting, and controlling parts.

5. Robots will monitor electronic operations including data bases and alerting users to changes. It will archive data and retrieve it upon request, supply it as a "consultant", or combine and filter it.

EXPERT SYSTEMS

1. "Corporate memory" will be increased by replicating human reasoning and expanding the size and complexity of the knowledge base.

2. ES will be used as decision maker for C3 battlefield applications. High volumes of data will require computers that offer higher processing rates. A system could evaluate a battlefield for traces of radiation and then perform a second task depending on the results of the first task and so on. These systems will need parallel processors to simultaneously cross-check results.

3. Systems will understand rules, data, and consider preferences in making decisions. These systems will be an electronic assistant or colleague problem solver.

4. Pilot aid systems will perform flight management, situation assessment, mission management, and hands-off flight. Some in-flight emergency systems are already being developed by Texas Instruments for the F14 aircraft.

5. Satellite stationkeeping tasks and on-board engineer maintenance functions will be performed automatically to ensure extended system life and enhanced mission capabilities.

6. Professional workstations will support tasks in the following areas: master schedule, public relations, leadership, liaison, negotiating, information processing, information dissemination, planning, controlling, resource allocation, marketing, selling, office automation and office operations.

IMAGE PROCESSING

1. Image understanding systems will develop graphics from photographs or relational data bases and determine targets of interest.

2. Satellite and aerial interpretation will identify objects in complicated patterns and backgrounds, convert data to grid coordinates and pass information to the battlefield commander.

3. Robotics will "see" 3-D in real-time. Large computer systems with parallel processing are needed to resolve the "waiting" issue.

4. Ground travel robots will have terrain vision and pathfinding capabilities that can see holes, trenches and other obstacles to determine the best path to travel.

FIFTH GENERATION MACHINES

1. General purpose machines will execute parallel software algorithms to provide flexible architecture for accessing information.

2. Very powerful work stations will allow a human in the loop for solving complex engineering situations and producing innovative programs.

3. General purpose machines will reduce software development time and in-

crease the range of applications. These machines will be small, fast, and very versatile for solving AI problems.

4. Powerful machines will process continuous speech understanding and provide continuous information retrieval. These highly parallel machines will support real-time applications in avionics, robotics, weapons, and other C3 areas.

5. Very Large Scale Integrated Circuits will allow high parallel processing on small chips that are very hard electronically and require little power. When placed in series, powerful computing processors will be available that can operate in any hostile environment including space or underwater.

TRAINING

1. Workstations will be available that will teach skills that are difficult to learn. Concepts underlying how people think and learn will be understood to improve learning curves.

2. Systems will transfer large amounts of data from an instructor and train managers, sales people, operators, lawyers, doctors, and almost anyone else. These will be known as trainee's assistant.

3. Systems will teach military personnel all aspects of equipment maintenance and basic skills including fire-arms training. These will be known as trainer's assistant.

SECTION 4
TECHNOLOGY AREAS

Much of the interview process was devoted to the question of new AI technology and what new cost methods will be required to estimate those technologies.

When making a cost estimate of a proposed system, the cost analyst has to make sure that the cost estimating tools have been developed for the estimating problem. These estimating problems will differ with the degree of technology advance required in the system as a whole or in specific subsystems.

Technology advance can be classified into three different levels. State-of-the-art (SOA) requirements apply to subsystems which have been built or are so close to subsystems which have been previously built that no advance in technology is required. Here, our traditional cost estimating tools are extremely good even for AI applications. Analogies will usually exist and CERs built for interpolation are already available.

The second technology level, called evolution, refers to extensions of existing technology which will normally happen given enough time. They do not depend on any great technical breakthrough. What is needed for cost estimating here, are techniques already developed that can be extrapolated in the required direction of technology growth. Some existing estimating techniques (CERs) will hold up well when tested for extrapolation, others will not.

The third level of technology growth is one that requires a technical breakthrough. What is desperately needed is an acceptable technique for examining the system physical/performance characteristics to quickly identify a breakthrough and its cost impact. This section will focus on AI technology areas that the cost estimator must anticipate and then prepare new tools or modify existing tools to predict system performance and cost. The technology areas discussed are:

T1. VHSIC
T2. Problem Solving Techniques
T3. Knowledge Representation
T4. AI Languages
T5. Commercial Tools
T6. System Specifications
T7. Development Methodology
T8. Schedule/Cost Estimating
T9. Technology Roadmap

Also mentioned in interviews was the Ada computer language. There could be an AI technology impact from Ada, but within the AI community, there is no wide interest in the subject. Also, the development and use of roadmaps for technology and system requirements has broad interest and is therefore included as a technology aiea.

Cross referencing is made throughout the report. A notation was adopted to facilitate cross referencing where technology areas in this section are labeled with a T, (i.e. T1, T2...). Research areas in section five are labeled with an R (i.e. R1, R2...).

TECHNOLOGY AREA T1
VERY HIGH-SPEED INTEGRATED CIRCUITS (VHSIC)

DESCRIPTION: Estimate the VHSIC cost impact when integrated into AI systems.

SIGNIFICANCE: The VHSIC program has provided great impetus to the research, design, and production of devices that will be the eyes and ears of surveillance and C3I expert systems. Significant manufacturing changes are required to produce devices that can solve hardness problems, and meet the tremendous speed and throughput requirements of DoD systems.

DISCUSSION: Individual processing capacities of hundreds of LIPS (Logical Inferences Per Second) are foreseen as AI requirements that are achievable only through VHSIC technology. Inference Corporation has estimated that hardware architecture requirements way in excess of this will be needed to process hundreds of parallel paths at the same time.

Estimates on AI costing need to be made. Will cost reductions due to miniaturization continue? Will the VHSIC technology enable the development of the tremendous processing power required for AI systems?

VHSIC offers speed, miniaturization, size, and density advantages by an order of magnitude which are all important to large AI applications. VHSIC production is yielding higher production rates needed for expert systems that process radar signals, communication, electronic counter measures, and missile control.

RELATED AREAS: Gallium Arsenide technology may some day replace VHSIC. Other new technologies on the horizon are Microwave/millimeter wave Monolithic Integrated Circuits (MIMIC) analog circuitry. VHSIC technology impacts cost models (R1), cost data base (R2) and interface costs (R12).

DOABILITY: The ability to build an AI system in silicon will be possible that will move us from the present 1 LIPS to an operations rate of thousands of LIPS. A basic ground rule is if a function can be built on a single board today, it can be fabricated on a single chip in a decade. If that function can be built in a one cubic foot box today, it can be fabricated on a chip in 20 years.

PRIORITY: High

RECOMMENDATION: VHSIC technology must be studied to anticipate cost impact on system efficiency and application performance.

RESEARCH STEPS:

1. Conduct a research study to expand COCOMO (R1). When VHSIC production begins in 1987-1988, this availability of processing power will increase program sizes beyond the current cost model capabilities. A microelectronics roadmap should be sponsored.

2. Conduct a research study to develop a cost data base (R2) as VHSIC processing production will decrease the hardware cost of computer systems.
3. Conduct a research study of interface costs (R12) where VHSIC capabilities will increase the number of subsystem applications that can be packed in each assembly. Each circuit will be capable of processing additional software or hardware interfaces.
4. Obtain results from ESD VHSIC Technology Transition Plan concerning capabilities, cost, and availability of VHSIC Phase I chips.
5. Perform studies that evaluate cost/performance benefits associated with VHSIC chips.
6. Conduct VHSIC Technology Roadmap to evaluate cost/schedule impacts of applying new VHSIC capabilities.
7. Track development costs of on-going projects when VHSIC capabilities are inserted.

TECHNOLOGY AREA T2
PROBLEM SOLVING TECHNIQUES

DESCRIPTION: Estimate the cost impacts associated with various methods of AI problem solving techniques. AI is a field concerned with problem solving in general and is specifically focused on duplicating the behavior of the expert solving the problem.

SIGNIFICANCE: Alternate problem solving techniques and possibilities greatly affect the cost and capability of an AI system. The AI knowledge engineer needs to know what information to process, how to encode it, and which technique is best to process that information.

DISCUSSION: There are two parts of an expert system namely the knowledge base and the inference engine. This section discusses the inference engine and how it can control, guide, and use the rules and data stored in the knowledge base to solve problems. Technology Area T3 discusses how those rules and data may be stored. Inferencing addresses the problem how to use the rules or how new facts are derived from known facts. It is accomplished three ways:

1. Modus ponens is the inferencing technique which states if a rule is true we may believe the conclusion. An example is that A is true and the rule is "If A then B", we can believe B.

2. Reasoning about uncertainty is the inferencing technique that attempts to consult or advise with information missing. Just because not all the information is available, it may still be necessary to make decisions.

3. Resolution is the inferencing technique to discover if a new fact is valid given the current knowledge base or present set of logical statements. This is the best technique for performing C3 applications when counter-intelligence data or counter measure activity are encountered because each piece of data is given a "sanity check."

Control methodology addresses the problem of where to start processing the knowledge base and how to resolve conflicts when alternate logic paths occur. It is accomplished by:

1. Backward and forward chaining. Backward chaining is goal-directed that reduces current application goals into easier, simpler to achieve sub-goals. Forward chaining is data-directed for responding to unanticipated situations and for solving problems where the form of the solution is not well understood.

2. Depth-First versus Breadth-First Search. Depth-First searches for detail first much like a specialist that would focus on a particular aspect of the problem. It probes all the rules and data digging deeper and deeper into detail. Breadth-First search sweeps across all premises, and inquires in a general way about the problem aspects before digging deeper.

3. Monotonic vs. Nonmonotonic Reasoning. Monotonic reasoning states that all values and facts that become true remain true so that the volume of system information grows. Nonmonotonic reasoning states that facts that are true may be retracted so that as new information is gathered, all data is re-evaluated.

Each inference and control technique has trade-offs that are not entirely understood technically and certainly the cost impacts of these trade-offs is less understood. Each has a loyal following that is willing to battle holy wars and always for a good explanation. A solid understanding of when to use which technique for a particular solution should be studied. A system development cost can be greatly affected using an inappropriate, inefficient, inaccurate or invalid problem solving methodology.

RELATED AREAS: The data base structure drives the control and inference technique used. Technology area T3 will address this. Cost research drivers impacted would include cost to integrate (R5), cost to maintain (R6), and productivity rates of development team (R9).

DOABILITY: This would be a moderate effort with a nice payoff. Some data and knowledge already exist on system cost differences using the different problem solving techniques. More complex strategies to problem solving will become available soon, and those techniques will require additional cost studies.

PRIORITY: Extreme

RECOMMENDATION: Inference and control are actually integrated throughout an AI system. The cost study of these techniques would provide a good performance CER and a good first task to begin AI costing.

RESEARCH STEPS:

1. Select systems and identify control and inference strategies used.
2. Study program size, programmer learning time, and maintenance cost incurred with the different strategies.
3. Perform study and correlate AI application to inference and control technique used.
4. Develop cost and technology trade-off of each technique.
5. Develop a CER based on number of rules.
6. Develop a CER based on control methodology.
7. Develop a CER based on inference technique.

TECHNOLOGY AREA T3
KNOWLEDGE REPRESENTATION

DESCRIPTION: Develop a cost estimating relationship based on the knowledge base structure and how it affects the software processing techniques.

SIGNIFICANCE: Knowledge base systems utilize symbolic data that must be organized to best accommodate the processing control and inference techniques used by the inference engine. In a worst case scenario, if the knowledge strategy doesn't support the control strategy, no conclusion or even worse an invalid conclusion would be the result.

DISCUSSION: There are presently five different techniques to encode the facts and relationships that represent knowledge. Each method has advantages and disadvantages.

1. Semantic Networks is the oldest and most general representation scheme. A network of objects called nodes are connected together by links. Flexibility is the major advantage with this scheme but it doesn't handle knowledge exceptions well.

2. Object-Attribute-Value Triplets is the scheme used in MYCIN where objects may be physical objects (a door) or a conceptual object (a logic gate). Attributes are assigned (size, shape, logic setting) where various values specify the nature of the attribute. The advantages to this scheme are data may be dynamic, objects may be related, and uncertain facts may be represented.

3. Rules is the scheme used to represent relationships that may describe uncertainty or pattern matching variables. This scheme requires complex rule look-up tables.

4. Frames is the scheme used to describe an object with slots of information containing all the attributes. Frames allow a rich representation of knowledge but are complex and difficult to develop.

5. Logical Expressions is the scheme where propositions are statements such as True, False, And, Or, Not, Implies. This provides theoretical elegance descriptors of knowledge (i.e. complex relationships between and among many different sets of data) but doesn't allow direct knowledge searches.

Different tools, different control and inference techniques, and different AI program languages all require different formats of knowledge representation to provide consistent results quickly. These interactions are not fully understood presently and yet are the high impact items for an AI cost estimating relationship.

RELATED AREAS: Knowledge representation is one of the major drivers to problem solving techniques (T2) and development methodology (T6). How to capture knowledge is discussed in research area R10 and the Costs of Integrating and Maintaining systems is discussed in research areas R5 and R6 respectively.

DOABILITY: Again, a modest study effort would produce a high payoff. Correlating knowledge representation techniques and control techniques could produce important insights to productivity and costing when developing AI systems.

PRIORITY: Extreme

RECOMMENDATIONS: Study the various knowledge representation techniques and how they impact the cost of developing AI systems.

RESEARCH STEPS:

1. Select a variety of systems that utilize the various knowledge representation techniques.
2. Normalize system sizes and determine development times.
3. Study other knowledge representation relationships such as knowledge engineer learning curves, cost of system integration and maintenance, and expert satisfaction with processing speeds and results.
4. Map system application to knowledge representation techniques for analagous projects.

TECHNOLOGY AREA T4
AI LANGUAGES

DESCRIPTION: In the previous Technology Areas, the potential new computers, knowledge representation techniques, and control techniques were provided and the importance of these technologies as they affect AI systems discussed. This section describes the AI language requirement and T5 will discuss the need for additional development tools.

SIGNIFICANCE: The techniques of knowledge representation and control permeate every area of an AI system and it is the capabilities of the AI language that compile the knowledge and execute the control to solve the problem.

DISCUSSION: Programs may be written in machine language or at a higher level using C, Cobol, Fortran, Ada, etc. At an even higher level are AI languages such as Lisp, Prolog, OPS5, Emycin, and Interlisp. First generation AI languages each support a single technique for knowledge such as forward-chaining or backward chaining. Each is effective for certain classes of applications but no single technique has been found successful for all knowledge based systems.

Research is being performed to develop rule-based programming languages that may integrate some or all of these data representation, programming, and program control techniques into a single language architecture. This language would be to AI what many hope Ada will be for conventional programs. This very high level, single syntax, general purpose language would profoundly affect the cost to design, develop and maintain AI systems. This is a technology area well beyond the SOA, and when implemented, new tools will be necessary for the cost estimator to measure the rate and direction of technology growth and the resulting decrease in AI development costs. A universally accepted and used language would require software CERs for AI to be modified the same as Ada will require conventional software CERs to be modified. The cost estimator will need to stay alert for any new languages that would require different estimating techniques for the new technology.

RELATED AREAS: Current capabilities and cost implications of languages presently available needs to be studied as indicated in AI Languages Proficiencies (R13).

DOABILITY: A new, integrated AI language is still an R&D, academic concept that has a low probability of any major technology breakthrough in the near future.

PRIORITY: Low

RECOMMENDATION: Perform study about current language capability (R13).

RESEARCH STEPS: None at this time.

TECHNOLOGY AREA T5
COMMERCIAL TOOLS

DESCRIPTION: Estimate the AI system development cost as more powerful commercial tools become available.

SIGNIFICANCE: As tools become more powerful and easier to use, the cost of building and maintaining AI systems will be greatly reduced. According to the majority of experts interviewed, this is the primary AI cost factor.

DISCUSSION: This technology is the most important area that will impact the AI technology and its cost. A tool features: choice of knowledge representation, inference engine, a High Order Language (HOL) for specifying the knowledge base, all of the man-machine-interface packages, and the program development and debug packages. This is the environment that creates the AI system.

Tools come in three sizes:

1. Small system building tools that can run on PC's and generally are used to develop systems of less then 400 rules such as ES/P ADVISOR, Expert Ease, INSIGHT, M.1, Personal Consultant, Series-PC and K:base.

2. Large, narrow system building tools that run on LISP computers (or larger) and are used to develop a single system of 500 to several thousand rules that solves one problem domain such as EXPERT, KES, OPS 5, S.1 and TIMM.

3. Large, hybrid system building tools that run on LISP computers (or larger) and are used to develop many systems of 500 to several thousand rules each such as ART, KEE, SRL+ and LOOPS.

The availability of tools will ease the building and maintaining of AI systems. The developer and end user both benefit because tools are: simple to use, available on a variety of computers, can be applied to any domain of knowledge, provide a choice of inferencing techniques, provide a convenient method to capture knowledge, and provide a complete support and training environment.

As each new tool is implemented and used, the cost estimator must be aware that cost techniques may need to be modified. A new tool could be powerful enough to revolutionize how easily AI systems are implemented and maintained. This tool breakthrough could eliminate a current technical barrier to the degree that development costs would be drastically reduced. The cost estimator needs to stay alert to evolving tool technology and update cost relationships as new tools become available. Technology roadmaps will be used to determine tool SOA advances and resulting affects on system characteristics and development costs.

RELATED AREAS: Tools can impact AI knowledge representation techniques (T3) and increase productivity rates of various AI languages (T4). Present tool capability and productivity are discussed in tools (R14).

DOABILITY: Large amounts of industry R&D money is invested in the development of tools that include KEE, KES, and ART. Engineer workstations such as SUN and IBM presently account for a $3 billion a year industry.

PRIORITY: Extreme

RECOMMENDATION: This is the biggest payoff area for production rates and AI system costs.

RESEARCH STEPS:

1. Review all vendor literature about new tools already released and those soon to be released.
2. Interview vendors for new technology areas that tools could be utilized.
3. Develop technology roadmap including possible tool insertion date.
4. Develop SOA performance characteristics and the anticipated characteristics when tool becomes available.
5. Select current systems and identify costs for SOA performance characteristics.
6. Develop new cost estimating relationships and analyze technology impact on system performance and cost.
7. Update technology roadmap regularly.

TECHNOLOGY AREA T6
SYSTEM SPECIFICATIONS

DESCRIPTION: Estimate the cost impact of the customer writing rigid system specifications so the developer can provide a cost estimate, the programmer can build the application, and the tester can verify the results.

SIGNIFICANCE: The lack of system specifications makes the notions of a credible cost estimate or a formal verification proof absolutely meaningless. And yet, the very problems that AI systems resolve can not be formally documented because they process changing requirements using ambiguous and intuitive data.

DISCUSSION: Virtually all conventional programming methodology is based on specifications supplied by the customer and still implementation disasters occur with overruns in both time and cost. Any attempt to obtain exact specifications of an AI product is bound to fail because the customer does not really know what is wanted or cannot anticipate exactly what is required. By consensus of developers interviewed, AI systems are developed by prototyping then demonstrating to the customer then modifying and demonstrating again by continuous evolution until the customer either accepts the product or runs out of money. Without the specification agreed upon between customer and developer, the customer is never satisfied with what is demonstrated, the developer can not accurately estimate how long it will take to develop, and the development manager can not accurately estimate how much it will cost.

The first crucial area of software development (AI or conventional) is planning: the developer needs to know what functions the software is to perform and the various situations that it must process. Specifications, or deciding what action must be taken and at what point in time, becomes more difficult as the task to be solved becomes more complex. Simply writing out all of the situations to be processed and the appropriate actions for a C3 system would be a crude specification that could require thousands of pages. The better approach would be to modify existing specification requirements to include a method to define AI projects.

Conventional programming specifications are described in detail in MIL-STD-2167 and, it is agreed, that some AI systems do not lend themselves to the A Spec, B Spec, PDD, PDS, ICD, DBDD specifications. It is also agreed that some of these specifications do apply to AI as defined or if modified. Expanding the conventional specification techniques to AI specification techniques will provide technical descriptions, that lead to baselining that in turn will lead to bounding the cost of an AI product. Providing systems specifications will be a first step to determine any software CER's.

RELATED AREAS: Just as formal specifications are necessary for cost purposes, so is a formal development methodology necessary as discussed in technology area T7. Expanding Cost Models (R1 and R2) will need this data before any cost analysis is meaningful.

DOABILITY: RADC Branch COE presently has a task to develop a tool that will perform analysis of design rules and requirements to generate a complete set of AI specifications.

PRIORITY: Extreme

RECOMMENDATION: Begin a study project to map conventional specification methods to AI development techniques. This would be a short term high payoff impact on cost estimating because development costs can be accounted once the customer and developer agree on what is to be implemented.

RESEARCH STEPS:

1. Analyze results of RADC-COE studies concerning AI specification techniques.
2. Track implementation costs of AI system by specification/program module/AI function.
3. Track costs of unexpected changes.
4. Map AI system development techniques to MIL-STD-2167.
5. Map AI functions such as knowledge base to current specifications such as Data Base Design Document.
6. Map AI capabilities to conventional test techniques and QA methods.

TECHNOLOGY AREA T7
DEVELOPMENT METHODOLOGY

DESCRIPTION: An AI design and development methodology is necessary to cost estimate a system, manage a development team, and evaluate its progress.

SIGNIFICANCE: Without an AI development methodology where a product is subdivided into some organization of program functions or development phases, cost tracking for historical patterns and cost estimating for analogous products is impossible.

DISCUSSION: At least two people are needed to develop an expert system: the AI scientist and the expert. In the beginning, the AI scientist designs the knowledge base structure, the inference strategy, and the basic user interface. He acquires knowledge, by trial and error, from the expert and is then able to construct a first approximation of a working system by modeling a portion of the problem that seems most accessible to early solution. This prototype provides the reference point for later work.

As new aspects of the problem are introduced, the prototype is modified and tested. The prototype is expanded and refined until one of two events occurs. Either the expert concludes the system can solve problems with a high degree of expertise or the customer budget is exhausted.

AI is still in its infancy and has not developed a scientific development approach to replace the iterative, rapid prototyping method. More importantly, the necessary technology to associate costs to AI development phase are fuzzy. Present AI cost estimating is performed using the iterative process also. At project start date, the customer is briefed on a first-pass, "ball park" cost figure. As the project progresses, the cost estimate is refined bi-monthly where each estimate has a higher confidence of accuracy.

Just as conventional software is developed and costed using a waterfall phased plan, so AI projects need an equivalent development and cost plan. DARPA has attempted a two phase approach: phase one performs analysis and is fixed price; and phase two performs development and is cost plus. Inference Corp. uses a three phase concept: prototype, pilot, then development. Arthur D. Little uses a six phase development concept. Since most AI is developed with R&D funds though, project development normally continues as long as money is available and no cost accounting is performed. It is extremely important to determine an AI development methodology, track the associated costs of each phase, and then determine an AI cost model.

RELATED AREAS: The form and content of system specifications (T6) are directly dependent on the form and content of the development phases. These development phases impact present cost models (R1), cost data base (R2), schedule estimating (R3), and program management (R16).

DOABILITY: Many AI projects are developed, managed, and cost estimated by "winging it". The pay off for developing a methodical, engineering approach to AI programming is critical because costs can be tracked, schedules planned, and manpower estimates predicted.

PRIORITY: Extreme.

RECOMMENDATION: Task an independent contractor, STARS or SDI committee to define an AI development approach.

RESEARCH AREAS:

1. Perform specification study (T6).
2. Establish applicability of conventional software development techniques.
3. Compare AI technique and define a phase approach concept.
4. Develop definitions that include WBS of each increment, module interfaces, end products, organization structure, and a QA plan.
5. Identify successful AI projects.
6. Analyze project and determine WBS.
7. Allocate development cost to WBS.
8. Analyze for cost/WBS CER.
9. Develop data base.

TECHNOLOGY AREA T8
SCHEDULE/COST ESTIMATING

DESCRIPTION: Develop techniques to estimate AI project schedule, necessary manpower requirements, and resulting development cost.

SIGNIFICANCE: Software now represents 50% of a system development cost and up to 90% of a system life cycle cost.

DISCUSSION: Cost estimating for AI software is presently done by "winging it." By consensus, all AI developers, managers, and government personnel agreed that the need to competently predict AI software cost estimates is of paramount importance but not one person has a good, reliable method to do that.

AI software development technique is very different from conventional software development but many conventional CERs are valid for AI. Of primary concern to all AI experts is that the work performed up to date developing software cost models be maintained as the basis for AI cost estimating. Modifying present cost models is the least risk and recommended approach for developing a new AI cost model but to analyze data and then modify present models is a sizable task. Many cost variables are dependent on technical capabilities beyond the AI SOA. A technology roadmap to determine SOA advances is necessary in order that the cost estimator be able to anticipate and plan for updating those cost variables.

Many ideas and recommendations were presented during interviews about what parameters should be considered in an AI cost/schedule estimating model. Those ideas are presented below:

1. The impact of new technology and new research is the topic of this paper and a major concern to all interviewed. Measuring the impact of these areas on AI programmer productivity will be needed as each area is developed and studied.

2. Software size (number of SLOC, frames, or rules) is the single most important cost driver in both current and AI estimating models. Initial attempts to model AI efforts use a rule or frame count CER but presently there is no industry accepted agreement concerning what is a rule or how to count it.

3. AI languages are the next most important cost driver for AI models. Different languages vary in complexity and capability. Complexity affects the programmer learning curve and productivity curve. As languages vary in capability, rule complexity and length vary which affects rule count. A program written in 1000 rules may be written in only 100 rules using a different language (SLOC expansion factors). A study is needed to understand these measurements and how they impact cost/schedule estimates.

4. Verification and validation of AI projects is a major effort and needs methods for cost estimating.

5 Host computer cost for AI systems is still very high and in many cases, special purpose AI machines are required. Cost relationship for CPU time, execution speed, and mainframe expense are necessary.

6. System security (especially for SDI) is pervasive in all segments of software, hardware, and communications. Automatic vs. interactive security methodologies, number of security layers and levels required, number of system interfaces, and number of system processors all impact security cost considerations.

7. Diseconomy of scale in AI systems is a major cost impact. Development costs are not linear. Larger systems cost proportionately more to develop than smaller systems.

RELATED AREAS: Modifying present cost models (R1), developing a technology roadmap (T9), and using an AI cost data base (R2) are needed to support cost and schedule estimating affected by new technology.

DOABILITY: Some work has been previously done on estimating AI software schedule and cost but results and analysis are still very tentative. More data is needed before any quality CERs can be calculated.

PRIORITY: High

RECOMMENDATION: Evaluate present software cost models and extrapolate for AI considering new research data and new technology factors.

RESEARCH STEPS:

1. Develop technology roadmap.
2. Determine system performance characteristics of implemented AI projects.
3. Evaluate successful systems and develop system performance CERs.
4. As technology becomes available, evaluate CERs for new or modified performance characteristics.
5. Modify cost models for new or modified CER and update data base.
6. Identify the factors that influence the cost and schedule of AI implementations.
7. Identify the factors that influence the cost and schedule of improvements to the AI methodology.

TECHNOLOGY AREA T9
TECHNOLOGY ROADMAP

DESCRIPTION: Develop an AI technology roadmap so the cost community can focus on studies and research efforts to prepare for estimating new technology cost impacts.

SIGNIFICANCE: Present estimates indicate the cost to develop and maintain software represent 90% of a total system life cycle costs. New technology could greatly modify that percentage number.

DISCUSSION: As new AI technologies and tools are available, AI cost estimating methods will be modified. An in-depth identification of emerging technology is not the focus here, but as technologies are developed, cost and performance curves will change. During the interview process, many ideas were presented and are listed below:

1. Need to develop AI software and performance CERs that provide inputs that are recognizable to current cost models. As 4th and 5th generation systems become available, CERs must be determined as inputs to 3rd generation cost models.

2. Need to develop a method to write precise, well defined, testable, AI system requirements and specifications.

3. Need to model AI inference technique for cost estimating. Model must include number of predicates and rules, size of test engine, and number of interfaces.

4. Need to develop hardware cost curves. Computers are becoming cheaper because of VHSIC, GA, and 5th generation technology.

5. Need an AI shell that allows the expert to directly perform knowledge capture and develop system programs without help from the knowledge engineer.

6. Need a general purpose tool that can automatically develop and maintain the knowledge base.

7. Need to develop V&V specification for AI projects.

8. Need to perform AI logic processing using incomplete or uncertain data.

9. Need a tool to provide appropriate explanation of system behavior and conclusion.

10. Need to develop an AI procurement methodology.

11. Need a tool to evaluate rules and their results.

12. Need a tool to evaluate and predict logic paths executed during program run.

13. Need to develop the capability to search data backward or search non-chronological data.

14. Need to develop the capability to process 3-D visual data and discriminate edges and shadows.

15. Need to develop tools so that the expert can write programs directly.

16. Need the capability to store and process multi-logic states using analog computers.

17. Need a tool to automatically verify AI systems.

18. Need to develop AI industry standards, practices, and procedures for all life cycle phases of AI projects.

19. Need to teach customers an appreciation of the AI development technology.

20. Need capability to solve different requirements/problem with one general purpose AI system.

21. Need the ability to solve problems using evidential or probabilistic reasoning.

22. Need to develop a standardized industry definition for counting AI rules.

23. Need to develop AI software QA metrics.

24. Need the computer capability to process hundreds of parallel paths in realtime.

There is no universal consensus of the most critical technology insertion area but there is agreement that many areas need to be studied.

<u>RELATED AREAS:</u> This investigation should be done for any technology breakthrough with implications of system capability on current cost estimating techniques (R1).

<u>DOABILITY:</u> Cost models based on system performance can be updated as new technology is inserted.

<u>PRIORITY:</u> High

<u>RECOMMENDATIONS:</u> Study technologies and prioritize by technology insertion data.

<u>RESEARCH STEPS:</u>

1. Select technologies by priority that will be studied.
2. Select systems with existing performance/cost curve.
3. Track technology and determine insertion point in time.
4. Re-evaluate performance/cost curve.

SECTION 5
RESEARCH AREAS

Comments from interviews were reviewed and cross referenced into topic areas. Topics were studied to find problems of common interest, to find problems that impact AI system performance characteristics, or to find problems that require new cost estimating methods. For some topics, research has already begun, but AI is such a new and evolving technology that more significant data must be gathered before any valid conclusion about AI cost estimating can be made. This section discusses the following topics:

R1. Expanding Present Cost Models
R2. AI Cost Data Base
R3. Schedule Estimating
R4. Quality Estimating
R5. Integration Cost
R6. Maintenance Cost
R7. Security Cost
R8. Computer Trade-Off
R9. Level of Effort Cost
R10. Knowledge Capture Cost
R11. Inference Mechanisms
R12. Memory Management Cost
R13. AI Language Proficiencies
R14. Tools
R15. Personnel Considerations
R16. Program Management

Cross referencing is made throughout the report. A notation was used to facilitate the cross referencing where research areas in this section are labeled with an R (R1, R2...). Technology areas in Section 4 are labeled with a T (T1, T2...).

RESEARCH AREA R1
EXPANDING PRESENT COST MODELS

DESCRIPTION: Develop AI software cost estimating techniques building from conventional software cost estimating models such as COCOMO, PRICE, JENSEN, or DOTY.

SIGNIFICANCE: DoD anticipates its 1990 S/W budget to exceed $30 billion. The cost to develop and maintain S/W is a major system acquisition expense for most C3I, weapons, and avionics systems.

DISCUSSION: The commonly expressed fear within the costing community is that accepted, current S/W cost models could be discredited when applied to AI systems and not used at all. The commonly expressed opinion within the engineering community was that present AI cost estimate techniques produce, at best, only ballpark predictions.

The consensus was, in both communities, that continued research is necessary to develop new and better AI cost estimates. The recommended least risk approach is to decompose the AI effort into characteristics recognizable by COCOMO or others cost models and then use those characteristics as input parameters to those models. Those ideas are as follows:

1. AI cost estimates based on software size just as conventional software costs are based on source lines of code (SLOC). Cost analysts could convert the AI number of frames or number of rules into an equivalent SLOC count which in turn drives the COCOMO model.

2. AI cost estimates based on functional capabilities within the system. An AI system would be decomposed into a set of functions where each function is assigned an equivalent SLOC count based on analogy. The cost analysis would then use those SLOC values as inputs to COCOMO.

3. Cost estimates determined iteratively. A ballpark estimate is produced grass roots by interviewing the engineers that will actually perform the development task. As the system progresses, the cost estimate is reviewed and refined bi-monthly where each estimate is more refined and competent than the previous.

Other cost factors that need to be researched are as follows:

1. How to perform cost estimating of Verification and Validation (V&V). This effort is huge and in fact probably represents 60% of an AI development cost. Inference Corporation feels that testing an AI system is the same as testing any expert. Testing continues until either the system is accepted or discredited.

2. How to perform cost estimating of AI language experience. Different languages are different in complexity which affects programmer learning curves and productivity rates. Arthur D. Little, Inc. has found it takes 12 months of training for an experienced programmer to

perform as an apprentice AI practitioner and 60 months of training to perform as an expert that can work alone.

3. How to perform cost estimating of virtual machine experience. LISP machines are very complicated and hiring AI expertise from outside is impossible.

4. How to perform cost estimating of virtual machine volatility. All AI languages are being updated often which results in a major revision to the programming environment.

5. How to perform cost estimating associated with modern programming practices. As stated in technology areas T1, T6, T7, and T8, AI development practices and methodologies are still very informal.

6. How to perform cost estimating based on the availability and use of AI development tools. Some AI tools are still in the R&D phases and cannot be relied on to improve productivity in the near future. Many AI tools are available now or will be soon that will enhance programmer productivity rates and reduce development cost.

7. How to perform cost estimating for the expert's expertise.

8. How to perform cost estimating for the security required for the AI hardware, software and facility.

9. How to perform cost estimating for software systems that are larger than any system ever before developed. Current cost models predict to 1-2M SLOC sizes but some AI monoliths are expected to exceed 20M SLOC.

RELATED AREAS: Various related technology areas were referenced in the previous paragraphs.

DOABILITY: Some data already exists. Study of code characteristics, program environments, language complexity, etc. are all needed and would produce a large payoff for cost estimating.

PRIORITY: Extreme

RECOMMENDATION: Study available software development CERs and cost data sets to anticipate cost variations when producing AI projects.

RESEARCH STEPS:

1. Define current AI development approach.
2. Compare conventional and AI approaches.
3. Crystallize the differences.
4. Modify present cost models for the differences.

RESEARCH AREA R2
AI COST DATA BASE

DESCRIPTION: Develop an AI cost data base that includes all historical information describing AI development approaches, techniques, productivity, and costs. Establish a procedure for future data collection.

SIGNIFICANCE: The size and cost of all software is regularly underestimated and all cost models are driven by input parameters. Historical data must be organized into a data base so analysis can be performed to establish CERs.

DISCUSSION: Currently, cost estimating techniques do not accurately predict AI development costs. This limitation results from an inadequate understanding of the cost relationships of each variable and the scarcity of historical data. Cost models require historical data, identification of all the cost variables, and an understanding of how they relate.

A questionnaire needs to be designed to allow for the quantification of the development complexity and define the level of effort (LOE) for each development task. Some of the data sets needed include:

1. Size and cost of the host computer required for each system.

2. Labor mix and LOE of knowledge engineer, expert, and programmers during each development phase.

3. Final system cost.

4. Program size by frame or rule count for each phase.

5. LOE of prototype, pilot, development, V&V phases.

6. Program language used.

7. Tools used to design, build, and test system.

8. Number of interfaces, parallel processors, knowledge base facts, and development iterations required.

9. Knowledge representation, inference, and control techniques used.

10. Specificity of requirements documents and resulting deliverable documents.

11. Ease and accessibility of knowledge engineer and user to development graphics, tools, instruments, data, explanations and justifications, and prompt menus.

12. And all other data normally collected that is required by conventional software cost models.

RELATED AREAS: Technical areas T1, T5, T6 and T7 all require this data base for implementation. Developing an AI cost model (R1, R5, R6) depend on the data.

DOABILITY: Some data exists but an in-depth AI data collection survey is needed.

PRIORITY: Extreme

RECOMMENDATION: Conduct a task to identify, collect, collate, and relate AI cost drivers and variables.

RESEARCH STEPS:

1. Develop a list of cost drivers and variables.
2. Develop methodology to quantify and qualify each.
3. Develop a survey questionnaire.
4. Solicit AI industry for inputs and interview managers and engineers.
5. Collate data.
6. Modify 1 thru 3 to account for the responses in 4 and 5.

RESEARCH AREA R3
SCHEDULE ESTIMATING

DESCRIPTION: Develop a management tool and capability that aids in transforming the customer specification into a schedule plan and cost estimate.

SIGNIFICANCE: The impact of development schedule on system cost is significant. Schedule expansions and compressions increase cost.

DISCUSSION: Normally, software projects have overruns in cost and time. By admission, AI schedules are laid out depending on the budget and number of available personnel. Writing facetiously, this method is found to be 100% accurate as published by Harlow Mills. He found that engineers will always use the full amount of time allowed to develop their project.

An implicit assumption that the cost analyst makes is that the past is a good predictor for the future. This assumption is partly invalid for two reasons: cost drivers change as technology changes; and present managers learn from past management success and failure. The first priority to good schedule estimating is to collect consistent schedule data that is normalized for these two effects. Data to be analyzed must include:

1. Decreased time to build newer systems based on history where older systems provided lessons learned that in turn begat newer systems.

2. Decreased time to build recent systems based on the accumulation of ideas, code, and experience.

3. Decreased time to build recent systems based on the availability of workstations, languages, tools, decision aids, knowledge capture techniques, and computer development environment.

RELATED AREAS: Schedules are impacted by technology insertion mentioned in problem solving techniques (T2), development methodology (T7) and reinforces the need of a technology roadmap (T9).

DOABILITY: Schedule normalization from technology enhancement would be difficult but is needed. Present management schedule tools exist and could be modified once the schedule data is normalized.

PRIORITY: High

RECOMMENDATION: Collect historical AI schedule data. Initiate a task to normalize that data for AI technology advances.

RESEARCH STEPS:

1. Define what technology insertions affect schedule.
2. Obtain schedule data on systems developed before and after the technology insertion.

3. Compare development schedules of analogous systems for technology insertion impacts.
4. Build schedule estimating relationships.
5. Incorporate relationships into existing tools such as PERT Charts, Gantt Charts, WBS, and scheduling analyzers.

RESEARCH AREA R4
QUALITY ESTIMATING

DESCRIPTION: Develop a method to quantify and qualify software quality of the AI system.

SIGNIFICANCE: Quality of AI or conventional software is designed in rather then added-on. As a conventional software system moves through its development phases, it becomes less flexible to changes. In fact, to correct an error discovered during the requirements phase is 100 times cheaper to fix than that same error discovered in the O&M phase.

DISCUSSION: All of those interviewed expressed interest in this topic. This is understandable considering AI software development is accomplished by rapid prototyping which is a technique that does not follow the conventional software development methodology. It is difficult to measure performance, quality, and perform V&V for AI systems simply because these systems are developed incrementally and are constantly growing and evolving. The concerns of those interviewed were how to modify current QA techniques so that they could be applied to AI systems and what new QA tools and metrics are necessary to verify AI systems.

The need for QA tools that apply to AI systems was of immediate importance to developers. Some of the ideas discussed were as follows:

1. The need for a tool that can automatically test a system and verify the results. An interactive tool for SDI testing is much too slow. Presently, to verify software requires one test per 30 SLOC and that ratio computes at 300,000 to 1,500,000 tests to verify SDI. Only an automatic tool will be able to adequately perform that job in the time that will be allowed.

2. The need for a rule consistency checker that ensures correct and reasonable results are produced when different rules operate on the same data.

3. The need for a rule execution checker that ensures rules do not dangle or loop.

4. The need for a tool that can explain the logical process executed and why.

5. The need for a tool to debrief the expert during development. It must be able to explain which logic tree was executed that produced the solution.

The need for QA methodologies that apply to AI systems is important to avoid mistakes and predict error rates that ensure QA goals are achievable. Some of the ideas discussed are as follows:

1. Define AI development milestones and phases comparable to MIL-STD-2167. These events would mark the end of phases and provide a means to

track costs, review documents, and schedule review meetings.

2. Define a QA matrix for AI systems comparable to MIL-STD-2168. This would identify software qualities, define the form of the quality characteristic, and provide a means to determine if the quality requirement is satisfied.

3. Develop a data base used to predict error rates. Data would include project milestones, cost, quality, code efficiency, and code error rate.

4. Define quality attributes that are quantifiable and measurable and not just "high, medium, or low" value parameters.

5. Define a technique or review process to determine that a deployed systems meets it operational requirements during all phases of life cycle.

6. Define a V&V methodology for SDI that does not require a full system "live" test.

7. Define a test methodology that is quantifiable (system is sample tested not exhaustive tested). This avoids the approach of "test it until it's discredited or believed."

8. Define a configuration management technique that fits within the AI development scheme.

RELATED AREAS: The need for system specifications (T6) and development methodology (T7) areas directly applies to a QA capability. The cost to integrate (R5) and maintain AI programs (R6) is directly related on how much quality is built in during prototype and pilot development phases.

DOABILITY: A survey to quantify and qualify the quality of an AI product would be easy to perform and would receive industry wide enthusiasm.

PRIORITY: High

RECOMMENDATION: Perform a survey that analyzes what modifications are necessary to modify MIL-STD-2167 and MIL-STD-2168 to be applicable for AI systems.

RESEARCH STEPS:

1. Identify and define quality factors in AI products.
2. Assign impact weights to each factor.
3. Perform cost-to-implement vs. life cycle cost saving trade-off study.
4. Provide description of QA factor and how it relates to each application.
5. Identify tools required to ensure quality products and practices.
6. Develop these tools.
7. Document those practices.
8. Modify MIL-STD-2167 and MIL-STD-2168 for AI development techniques.
9. Develop a AI configuration management specification.

RESEARCH AREA R5
INTEGRATION COST

DESCRIPTION: Develop a model that predicts the cost to integrate AI systems.

SIGNIFICANCE: The cost to integrate a conventional software product into a system is tremendous. The integration cost of a conventional system is 160% the development cost and yet present cost models do not estimate this p.iase. There is a vital need to gather historical data for predicting analogous AI integration costs.

DISCUSSION: The classic software development method follows the requirements-analysis-design-implement-integrate-maintain model where all of the system integration test is performed in a all-at-once "big bang" approach. In this situation the software is not usable or functional until integrated. AI development method uses an incremental or iterative method of "build a little, test a little." In this situation, every increment is functional and each succeeding increment builds on the capabilities of the previous one.

Incremental development has a major advantage in that it supports an approach that allows an early evaluation of assumptions and risks. System refinement and test are embedded in the progress of development. This raises the issue then that there may not be a cost associated with the unit test phase of integrating an AI system.

System test is the second phase of integration that includes the cost normally associated with inter-system interfaces. The effort to interface an AI system and a conventional system is difficult to predict. AI products can produce unpredictable results that, if communicated, a conventional system would not be programmed to process. A first step to estimating AI integration costs, though, will be to use cost data from analogous conventional systems. This data should include number of interfaces, operating time restrictions, data rates, data formats, complexity of interface protocols and data.

RELATED AREAS: Simulation and other commercial tools (T5) would be useful during inter-system integration to recreate test situations. Integration cost data is stored in the cost data base (R2).

DOABILITY: Integration costs are normally well documented and available for analysis.

PRIORITY: Low

RECOMMENDATION: Perform data gathering task when developing cost data base specified in R2.

RESEARCH STEPS: Incorporate task under cost data base (R2) task.

RESEARCH AREA R6
MAINTENANCE COST

DESCRIPTION: Develop a model that predicts the cost to maintain AI systems.

SIGNIFICANCE: The maintenance cost of a conventional software system represents 80%-90% of the total life cycle costs.

DISCUSSION: Once an AI system is delivered to the customer, most systems grow, expand and require on-going programmer support. XCON was developed in 1980 for DEC to configure computer architectures. Initially, it analyzed 420 parts for the VAX 11/780 and reasoned using 750 rules. Today, XCON analyzes 8500 parts for many DEC computer systems and uses 4300 rules. The ratio of rules to parts shifted from 2:1 to 1:2 and requires 4 man years of effort each year to support it.

Life cycle support of AI systems is a problem because rules and the data base cannot be "fiddled with" by apprentice programmers. Programmer continuity and corporate memory are very important. A potential fix could be from COE Branch at Rome Air Development Center that has proposed a knowledge based software assistant. This is a R&D task that, if implemented, will identify critical expertise during each phase of the AI life cycle. During each phase, experts would save performance and technical expertise in an AI knowledge base. By the O&M phase then, corporate memory would be in the form of an expert system that could assist the maintenance programmer. In addition to this software maintenance tool, estimating maintenance cost for life cycle support becomes very important because knowledge bases and rules evolve over time. Data collection and cost analysis of current systems being maintained is required to predict and estimate these costs.

RELATED AREA: The information collected for the recommended study would be included in the cost data base (R2).

DOABILITY: Maintenance cost predicting is a challenging area and presently little data or technology exists to perform accurate estimates. A cost data base identifying AI maintenance costs is very achievable.

PRIORITY: Low

RECOMMENDATION: As more AI systems are developed during the next five years, the cost to maintain software will become available.

RESEARCH STEPS:

1. Identify AI system in O&M phase.
2. Determine system maintenance cost.
3. Develop system performance parameters and characteristics.
4. Develop system WBS.
5. Perform cost analysis and develop maintenance CER for WBS, system function, and system characteristic.

RESEARCH AREA R7
SECURITY COST

DESCRIPTION: Estimate the additional cost incurred when developing secured AI systems.

SIGNIFICANCE: SDI and many of the other anticipated AI systems will be classified and require secured software, facilities, and communications.

DISCUSSION: Rome Air Development Center, COTC Branch is very concerned about AI system security because security is pervasive and must protect all segments of SDI or any other system. COTC has awarded 6 study tasks that focus on various aspects of AI security including a guard/ward security concept, trusted/secure code development, SDI security verification for software specifications, high data rate security algorithm demonstration, security architecture for remote software modification, and a computer/communication security plan. The results of these tasks will improve DoD's understanding of AI security technology and may also identify some costs associated with implementing security capabilities.

Another concern includes the need for an AI knowledge tree that monitors security of inter-system communication including message release and route. An additional security concern is software system execution speed. An automatic security program would require system overhead and be too slow for a SDI realtime system but a semi-automatic security monitor would operate too slow to keep up with the data rates expected for SDI applications. And finally, there is concern about security verification. If AI systems can produce unpredictable results, what is the confidence that the AI security tree determined the correct classification or procedure?

Security cost data associated with conventional systems is available and makes a good basis for cost estimating. AI logic techniques will require new security techniques and may be a difficult costing problem. Cost estimates for the security of facilities and documentation would be analogous to existing systems. Security costing for AI will be a long term effort because as AI technology advances, its security requirements and specifications will probably also change.

RELATED AREAS: Research data would be included in R2 cost data base.

DOABILITY: This is a long term effort and will require an AI security system to be implemented before any cost predictions would be competent.

PRIORITY: Medium

RECOMMENDATION: Perform data collection of the security costs incurred when developing systems already operational and prepare a security cost CER.

RESEARCH STEPS:

1. Analyze current systems to determine portion of system cost associated with security cost.

2. Crude CER would be to estimate security cost as this percentage.
3. Evaluate current systems and identify security costs for facilities and documentation.
4. Analyze current system by operational function and WBS to determine security costs for function or WBS element.
5. Develop security cost CER by system function, or WBS element.
6. Apply CER to analogous functions and WBS elements included in classified AI systems.

RESEARCH AREA R8
COMPUTER TRADE-OFF

DESCRIPTION: Develop a cost relationship of AI system specifications and the required computer to perform the processing.

SIGNIFICANCE: Several AI development projects have met very disappointing results. The developer did not know the computer memory requirements and just piled things in with the hope that they would work. Sometimes they would not.

DISCUSSION: Personal computers can host small expert systems, process non-realtime programs, and cost very little. LISP computers are at the other end of the size scale. They are special purpose, expensive, realtime, powerful processors. Many of the developers interviewed were concerned about their capability to pick and choose the right computer for the right job.

Research must evaluate AI processing requirements vs. the cost and capabilities of the computers available. Computer trade-off studies will become more important as application systems become more complex and diverse.

RELATED AREAS: The computer trade off-study must include: which AI languages it can process (T4 and R13), number of interfaces (R12), and the problem solving techniques it can support (T2). System memory requirements would be extrapolated from the cost data base (R2 and R11). The availability of tools (T5 and R14) that can run on each computer is a major factor also.

DOABILITY: Some of the data required for the trade-off study already exists The special computer requirements necessary to support special AI processing could be developed.

PRIORITY: Low

RECOMMENDATION: A small trade-off study would provide the data to estimate computer costs for anticipated AI projects. A subsequent study would not be necessary until VHSIC Phase II is complete.

RESEARCH STEPS:

1. Identify successful AI systems.
2. Gather data to define hardware type, cost, memory, performance, and capability requirements for each system.
3. Develop hardware CER for system memory, performance, and capability requirement.
4. Perform survey of industry developers and identify special processing requirements necessary to support special AI applications.
5. Develop a matrix that maps computer attribute to associated cost.
6. Develop map for each candidate computer.

RESEARCH AREA R9
LEVEL OF EFFORT COST

DESCRIPTION: To predict by phase the required manpower to develop AI systems.

SIGNIFICANCE: Direct labor cost is a fundamental parameter to all software cost estimating models.

DISCUSSION: AI systems are developed in three phases: prototype, pilot, and development. Each phase requires a different distribution of labor and a different level of effort. During the prototype phase one or two experts and one or two knowledge engineers produce the knowledge base. During the pilot phase, a larger team of engineers work with one or two experts to refine the prototype. During development, the engineer team could grow to as many as 100 programmers and testers.

Team size is restricted to five or six developers during the first two phases. The Institute for Defense Analysis learned that larger teams are not efficient because so much time is necessary for a prototyping team to discuss system design, analysis, and development methodologies with each other. A rapid prototype environment requires a small team to avoid miscommunication among its members and avoid schedule bottlenecks.

Team size can be imposed by a compressed schedule. If team sizes are increased to accommodate a strict schedule, the programmer productivity rates are decreased because more design decisions must be communicated to more programmers. Developed AI systems should be studied to evaluate team size, resulting productivity rates, phase schedules, and total development cost. Diseconomy of scale should be determined for each phase and used for future cost estimates and schedule plans.

RELATED AREA: Management must plan staff requirements and development schedules (R16). Personnel turnover (R15) rates are affected if employees remain on a project too long and become attracted to other programs.

DOABILITY: Fair pay-off. Some data is available for collection to analyze AI phase distribution of labor but the sample size is small. There are only several hundred AI systems developed to present and many were produced using poor cost accounting and inadequate record keeping of direct labor utilized.

PRIORITY: High

RECOMMENDATION: A LOE productivity CER ,ould be valuable tool for planning schedules, manpower requirements, and project budgets.

RESEARCH STEPS:

1. Identify AI systems where development phases are identifiable.
2. Collect manning and schedule data by each phase.
3. Identify results of each phase.
4. Identify productivity criteria for each phase.

5. Develop schedule, level of manning, productivity rate, and total cost relationships by phase.
6. Review study annually as new AI products are developed.
7. Produce cost accounting procedures for AI systems to track labor costs and overhead costs by phase.

RESEARCH AREA R10
KNOWLEDGE CAPTURE COST

DESCRIPTION: Develop a cost estimating model that predicts the project costs of the knowledge engineer capturing and acquiring the expert's knowledge and understanding.

SIGNIFICANCE: To develop an AI project requires considerable effort to capture the expert's knowledge.

DISCUSSION: At least two people are needed to create an expert system: an AI scientist and an expert. The expert is intimately familiar with the problem, for instance, he may be an F-18 pilot. The AI scientist is a senior scientist that has considerable experience implementing expert systems. He must interview the expert and understand the decision making process the expert uses to solve the problem. The AI scientist with this understanding then decides the knowledge base structure, inference strategy, and the user interface design. When the thinking process of the expert is programmed to couple with the knowledge base of facts and the processing rules, the expert system should be able to analyze information and solve problems.

During the knowledge acquisition phase, the knowledge engineer acquires, by trial and error, a working knowledge of the expert's understanding. The resulting model changes as new aspects of the problem are introduced, older portions of the program are modified, and the system is tested. This process is expanded and refined in detail and sophistication until either the expert concludes the system matches his own capability to solve problems or the customer runs out of money.

This approach has several potential flaws such as:

1. The difficulty of taking into account the role of human intuition and emotion in decision making.
2. The expert may not be available. Gaining knowledge from an expert can take a considerable amount of time and it may be the expert has too many commitments or is too expensive.
3. The expert is unable to communicate his ideas. Expertise can be very difficult to quantify and the expert may not understand how he arrives at the conclusions he does.
4. The expert is unwilling to communicate his ideas. It is important that the expert not see the development of the ES as a threat.
5. There is no expert. HEARSAY-II failed to meet speech recognition criteria because speech understanding is not well understood. Nobody consciously does the many tasks involved in speech understanding.
6. Knowledge may be spread across different contractors which implies proprietary limitations. A contractor would not want expertise stored in a knowledge base that would be accessible to a competitor supplying different expertise but to the same systems.

Cost data for the knowledge acquisition phase does not exist. This could be the most expensive development phase. A CER is required that defines total system cost to the cost to acquire knowledge including the AI scientist cost and the expert cost. Potential cost impacts of the six problem areas sited above need to be accounted for in modeling this CER.

RELATED AREAS: Which problem solving technique (T2) and which knowledge representation technique (T3) the AI scientist uses is determined by his understanding of how the expert solves the problem. The cost of the knowledge acquisition phase impacts: cost models (R1), cost data base (R2), scheduling process (R3), and program manager techniques (R16).

DOABILITY: This task would be accomplished in parallel with other tasks recommended in this paper. Historical data can be determined from successful projects.

PRIORITY: High

RECOMMENDATION: Questions about the cost to capture knowledge must be included in the cost data base (R2) survey.

RESEARCH STEPS:

1. Identify AI systems already implemented.
2. Gather cost of expert and AI scientist to develop knowledge base of rules and data by development phase.
3. Develop knowledge capture CER to total system cost.
4. Ensure cost data base questionnaire solicits knowledge capture cost data.
5. Evaluate and refine CER as new data becomes available.

RESEARCH AREA R11
MEMORY MANAGEMENT COST

DESCRIPTION: Develop a technique to cost system overhead required to support dynamic memory architecture required for expert systems.

SIGNIFICANCE: Memory requirements during ES processing may expand or decrease which affect system operating capability.

DISCUSSION: Expert Systems need access to any memory because the identity of the next piece of data to be processed is often predicted by the results of the last rule(s). An ES needs access to any memory necessary to solve the problem. A conventional memory management architecture, on the other hand, pre-identifies its memory architecture. System memory managers must be developed to ensure all memory is available at the rate the ES requires.

System overhead becomes processing expensive as more memory is requested and reserved because data sets and operating capabilities need to be swapped. Some semiconductor firms are developing memory management units that let a machine believe it has access to virtual memory and this may be a solution. In the meantime, a memory management unit cost vs. system overhead cost study is necessary to validate benefits proposed by semiconductor firms.

RELATED AREAS: VHSIC (T1) may increase processing speed and hardware memory size.

DOABILITY: Any major near term breakthrough in H/W capabilities that increase memory speed or accommodate virtual memory requirements is remote.

PRIORITY: Low

RECOMMENDATION: Perform research after VHSIC Phase II implementation.

RESEARCH STEPS: None at this time.

RESEARCH AREA R12
INTERFACE COST

DESCRIPTION: Develop a technique to estimate the cost of ES interfaces based on the type complexity, and number of each.

SIGNIFICANCE: Development and maintenance costs are directly proportionate to the system complexity. System complexity in turn is related to the capability of each interface and the number of interfaces.

DISCUSSION: An expert system architecture may include three classes of interface: knowledge engineer interface, user interface, and external system interface. The knowledge engineer (KE) interface provides the means to edit the knowledge base, use trace and probe development tools, and directly interact with the ES through graphics displays. The user interface provides the means to query the ES to receive explanations and justifications of results, provides a menu that displays data values and logic rules, provides a menu that the user can ask for help or issue commands. The external system interface allows the ES to monitor and control sensors, instruments, data bases, and other external expert systems.

There are a number of costs that the customer should consider when designing the interface architecture: system processing time necessary to control and handle each interface, user training cost to learn how to operate the system, development cost to program an interface capability, and life cycle cost to maintain it.

RELATED AREAS: Interface make or buy decisions must be made by management (R16) and are based on the availability of commercial products (T5 and R14). The cost to integrate (R5), maintain (R6), and secure (R7) each must be identified and stored in a cost data base (R2).

DOABILITY: Obtaining good data will be a problem until off-shelf capabilities are available.

PRIORITY: Medium

RECOMMENDATION: The industry cost survey should include questions about interface cost.

RESEARCH STEPS:

1. Perform survey and obtain catalog of commercial packages, their capabilities and costs.
2. Perform study of ES and isolate cost to implement interfaces.
3. Develop CER of total system cost to cost per interface.
4. Develop roadmap to include insertion points of new off-shelf interface programs and estimated costs.
5. Update industry survey to include questions about interface costs.

RESEARCH AREA R13
AI LANGUAGE PROFICIENCIES

DESCRIPTION: Develop a technique that defines which AI language is most proficient to solve a particular problem and produce a particular application.

SIGNIFICANCE: Different AI languages have different built-in features. The considerations that lead to the selection of a particular language must be understood to efficiently solve the problem and at the least cost.

DISCUSSION: The language-tool continuum as shown in Figure 4 is a simple way of classifying the various AI languages, environments, and tools. Languages are more flexible and more difficult to use and only well trained programmers can build systems with Lisp or Prolog. Tools (as will be discussed in R14) are not very flexible because major technical decisions and capabilities are already incorporated and cannot be modified but if a problem can be solved with tools, it can be developed quickly and by novice programmers. Environments are in-between languages and tools. They are packages of prewritten code useful for a particular programming task such as libraries of subroutines.

```
INSIGHT                                   ADA
TIMM                                     PASCAL
M.1      ART
S.1      LOOPS                    INTERLISP
EMYCIN   KEE           OPS 5       PROLOG          LISP
<------------------------------------------------->
TOOLS                 ENVIRONMENTS             HIGH-LEVEL
                                                LANGUAGES
```

LANGUAGE/TOOL CONTINUUM

Figure 4

AI languages have built in features that make it easier to build expert systems with them. They handle symbolic processing where conventional languages are designed to handle numerical operations. It is easier to program an expert system in Prolog or Lisp then use a conventional language like Pascal or Ada. By analogy, a physics concept can be explained with ordinary language using algebra but that is a tedious process. Instead a physics concept is conveniently explained with a physics vocabulary using calculus.

Tools may be implemented with AI or conventional languages. Lisp and Prolog, until recently, were not available for mainframe computers. To make it easy to run tools on a variety of mainframes, they were written in Fortran and Pascal and structured to interface with expert systems. As time progresses, most AI tools will be written in an AI language just as a physicist prefers calculus over algebra.

The operating system used in the ES is also of major concern. Conventional languages use conventional operating systems such as MS-DOS, EXEC-8, OS/360, TENEX and SCOPE. UNIX offers an advantage because it is easily transported from one machine to another. In fact, many ES tools are developed with UNIX for this reason. On the other hand, AI programs run slowly under conventional operating systems because of the inefficient way the conventional O/S translates into AI machine language. Computers are now being developed that use Lisp as an O/S and they are called Lisp machines.

Not all AI languages are created equal. Each has strengths and weaknesses that require complex trade-off decisions about which to close when developing complex and sophisticated programs. The two major AI languages are Lisp and Prolog and are compared as follows:

Lisp

- represents data in multi-level lists
- simple syntax
- easy to learn
- interactive
- easily customized and extended (flexibility)
- easy to build knowledge base
- lists can be nested
- easy to modify
- widely used
- automatic data storage
- very modular
- interpreted and compiled runs

Prolog

- based on concepts of formal logic
- concise problem description
- concise knowledge base descriptors
- chosen by Japan' for 5th generation
- can integrate with Lisp
- compact
- fast
- good for Natural Language
- uses references
- more advanced (elegant)
- good interpreter (fast)
- provides pattern match
- provides back tracking

A quick overview of other languages on Figure 5 are also summarized here and tools are discussed in R14. Interlisp is a version of Lisp that contains many prepackaged tools. Prolog and Ops 5 are less flexible than Lisp but more flexible then M.1 or EXPERT. KEE can represent knowledge in several ways but is difficult to use. S.1 is a narrow tool used to build diagnostic/consultation systems very quickly.

In short, being able to pick the right language with the correct operating system greatly affects the success of the project and its subsequent cost. As more languages, tools, and operating systems become available, the harder it will be to make the best choice.

RELATED AREAS: AI Languages (T4) indicated work is presently underway ,that if successful, will integrate many capabilities and produce one language able to support many different techniques and applications. AI tools selection and resulting cost impact is discussed in R14.

DOABILITY: Some of the language capabilities characteristics are now available and could be analyzed for strengths and weaknesses.

PRIORITY: High

RECOMMENDATION: A research project with a good payoff for expert system cost predicting.

RESEARCH STEPS:

1. Compile list of expert systems.
2. Survey vendors and determine system capabilities, knowledge base characteristic, control and inference techniques and operating system used.
3. Determine AI language used.
4. Evaluate language for ease to use and understand, productivity rates, and expert compatibility.
5. Map language to operating system and analyze system capabilities and customer satisfaction.

RESEARCH AREA R14
TOOLS

DESCRIPTION: Develop a technique that defines which tools are best suited to support expert systems development.

SIGNIFICANCE: Tools are designed to capture expertise or are useful in solving a problem. It is a waste of time to try and develop an expert system using the wrong tool.

DESCRIPTION: Figure 5 indicates where the better known AI languages, environments, and tools are placed on the language-tool continuum. Languages were discussed in R13 and this section will discuss tools, their availability, and their cost impact when developing systems.

Tools are designed to facilitate the rapid development of an ES and may incorporate strategies for representation, inference, control, or elementary constructs to model the problem. It usually will address a specific, narrow class of problems. Tools offer two advantages. First, tools provide rapid development using a large amount of already tested and debugged computer code. Secondly, tools provide specific techniques for handling knowledge representation, inference, control, and problem solving models.

The knowledge engineer needs to be sure that the tool chosen to help solve a problem is appropriate for the job. Many problems are being solved for which tools already exist and are available. If properly used, development costs are reduced because less manpower is required and a smaller team is needed to analyze and code the program.

The following categories for each tool need to be evaluated for their cost implications and capability trade-off:

1. Consultation. What class of problems is this tool designed to solve? Some tools plan, some diagnose, some prescribe, some coordinate, etc.

2. Representation, inference, and control. Tools vary in their capability to represent facts and relationships when drawing inferences about new data and for controlling the reasoning process.

3. User interface. Tools vary in their ability and capability to interact with the user. Tool features include explanation facilities, graphical display of reasoning process, and on-line help systems. Tools may be used by the developer to build, modify, debug, expand and evaluate knowledge base. Tools provide trace, and break facilities. Tools may be a library of cases that can be run to verify that the modifications to the knowledge base do not change previous results.

4. Applications. This catalogs examples of systems previously developed with this tool.

5. Support. This category includes description of reference manuals, training courses and other services that make a tool easier to use.

6. Price. This is the tool cost per unit or cost if purchased in quantity.

7. Host computer. This describes the computer or workstation the tool runs on.

RELATED AREAS: Problem solving techniques (T2) and Knowledge Representation(T3) were discussed and technical trade-offs evaluated earlier. The cost impact of anticipated tools was discussed in T5.

DOABILITY: A modest effort would gather the data that define current tools and their characteristics.

PRIORITY: High

RECOMMENDATION: Implement data base of tool capabilities and develop tool CER to systems development costs.

RESEARCH STEPS:

1. Develop tool survey questionnaire based on seven categories listed above.
2. Survey industry for tool data and implement tool data base.
3. Obtain cost data for systems developed without tools.
4. Obtain cost data for analogous system developed with tools.
5. Compare data obtained in steps 3 and 4 and produce tool value added component.
6. Perform trade-off study of tool cost and its value added or cost saved during development.
7. Survey industry for anticipated tools and availability dates.

RESEARCH AREA R15
PERSONNEL CONSIDERATIONS

DESCRIPTION: Develop estimating techniques to anticipate costs associated with AI programmer learning curves, corporate memory, hiring, and turnover.

SIGNIFICANCE: It is expensive to prepare programmers to do AI work and then to retain them once they are proficient.

DESCRIPTION: It takes time to learn how to do AI work and those training costs are very high. Getting started in AI takes longer and is more difficult then other computer sciences fields because there is an inordinate amount of information from many disciplines that needs to be acquired and assimilated. Arthur D. Little, Inc. has determined a learning schedule. Assuming a computer scientist with a broad background, no previous AI experience, and training full time, the number of months to reach AI competence are as follows:

SKILL	WORK AS APPRENTICE	WORK ON TEAMS	WORK ALONE
Knowledge acquisition	1	2	3
Knowledge representation	2	3	5
Knowledge Base design	3	6	12
AI programming	3	6	12
General AI theory	6	9	18
Expert AI practitioner	12	24	60
AI systems maintenance	1	6	6

Training the AI programmer is time consuming and difficult but training high quality, in-house staff is better than hiring. In fact, it is impossible to hire AI expertise because there are so few of them and the competition among corporations for these professionals is intense.

Once trained and the project is complete, AI programmers change companies. The turnover rate of AI programmers is almost twice the turnover rate in other computer fields. When the system has been developed, the expert and the AI programmer react differently. The expert, who is loyal to the company and understands the business, will remain and search for new applications for the expert system. The AI programmer instead, will be attracted to other projects, probably outside the company, to solve new technology problems in new application areas. This loss of corporate memory can be devastating if the need to maintain or expand the expert systems becomes necessary. Rome Air Development Center is studying the corporate memory problem. The expertise learned during the various stages of AI program development is saved in a second expert system that can then perform as a programmer assistant during maintenance. This capability is still a long way from being implemented.

In the meantime, understanding the cost to train a computer scientist to be an AI expert needs to be determined. Additionally, personnel managers and task managers must develop techniques to ensure company loyalty of the AI programmer to maintain corporate memory.

RELATED AREAS: The role of program and personnel management to minimize employee turnover is discussed in research area R16. Training time impacts schedule estimating (R3) and the experience level of the AI expert relates to project quality (R4).

DOABILITY: Data exists to determine training schedules and costs. Identifying what policy and procedure changes are required to retain AI programmers is a long term study. Computer assistant tools that supply and embellish corporate memory are still several years from implementation

PRIORITY: Medium

RECOMMENDATION: Study training costs now. Study how to modify personnel and management policy and procedure as budget allows.

RESEARCH STEPS:

1. Select systems that required training the programmers.
2. Isolate and identify training costs.
3. Develop CER based on training costs to total system cost and size.
4. Survey AI experts and evaluate retention needs and methods to achieve them.

RESEARCH AREA R16
PROGRAM MANAGEMENT

DESCRIPTION: Determine costs and benefits management must consider to: make/buy an expert system, estimate its development effort, and manage its implementation schedule.

SIGNIFICANCE: Significant amounts of money are spent to acquire expert systems. But it could be even more expensive if the system acquisition and development is poorly managed.

DESCRIPTION: Expert systems are expensive to develop. Costs and benefits for each ES must be identified. Costs must include the effort of both the expert and the AI engineer and any new hardware or software that needs to be acquired. Expected benefits include reduced operating costs, increased productivity, enhanced or new products or services. Once the manager is convinced to develop the system, a decision is required on how to start the effort.

The ES can be started using one of four approaches where each approach has a different risk, cost, schedule, and goal.

1. "Flying Start" where AI experts are hired in, Lisp machines with applications software are purchased, and the system is developed in-house without outside help.
2. "Buy and build" where consultants are retained, in-house staff is concurrently developed, Lisp machines with applications are purchased when staff is ready, and the system is eventually developed in-house.
3. "Build from within" where AI capability is built by training existing staff, Lisp machines and applications are purchased, and system is slowly developed.
4. "Test the waters" where AI capability is built by training one or two people, buy a Lisp machine when staff is competent, very slowly develop applications with R&D funds.

Once the development project is started, the manager must now monitor its progress and ensure its success. Conventional software development projects were developed with a waterfall technique and management has an accepted methodology to estimate those schedules and monitor progress. Expert systems on the other hand, are developed using a rapid prototyping technique that required new and different management estimating and monitoring tools. ES projects are developed in three, four or five development iterations where each version is an enhancement or enlargement of an earlier version. Historically, managers have managed these efforts by "winging it." A new, scientific management approach is required that mirrors the rapid prototyping development approach. Project management will require a plan-control-monitor-adjust model just as the ES developing programmer use a "build a little, test a little" model.

Successful expert systems require a new management model to plan and guide project development. New management tools, methods, policies, and procedures are necessary. A major research and study effort is absolutely necessary to determine the who, what, when, where, and why policies and procedures required during each ES development phase.

RELATED AREAS: All technologies discussed in Section Four, as they become available, will im⁻act ES schedules and must be anticipated by management. All research areas discussed in Section Five, when complete, will provide management data and tools to assist decision making.

DOABILITY: Some techniques to manage ES development projects exist and are used but none are acknowledged as "the" correct methodology. Sampling techniques to determine proven and best approach methods is not difficult.

PRIORITY: High

RECOMMENDATION: Investigate existing ES management tools and procedures. Coordinate these findings and develop an accepted "best approach, least risk" management technique.

RESEARCH STEPS:

1. Attend ES seminars and survey industry to identify present management tools and methods.
2. Rank each for effectiveness and accuracy.
3. Develop new schedule and estimating methods.

SECTION 6
RECOMMENDATIONS

OVERVIEW

"The cost research roadmap will define what, when, where, priority and cost of specific task to be undertaken in the next five years that will enable ESD to provide quality cost estimates for AI/ES projects." This study approached the problem by first analyzing technology areas that will impact AI system performance, efficiency, and cost. Second, this study identified research areas that will provide understanding of AI cost and schedule estimating. Each of these two areas is discussed as follows:

1. Technology Areas: Artificial Intelligence systems employ development methods and techniques that current cost estimating task do not evaluate or utilize. The cost analyst has to make sure that cost estimating tools will exist that can evaluate the system performance, efficiency, and cost as these technology areas are understood or available. Nine areas were identified and discussed in Section 4 that will impact AI system cost and each needs to be factored as cost parameters.

   Each technology area was evaluated for priority and doability and summarized in Table 6.1. Priority was assigned by how important it is that the cost estimator understand the impact each technology will have on system cost and efficiency. Priority is graded as extreme, high, medium, or low. Doability was assigned by how difficult it is to achieve success in understanding that technology area and its impact. Doability is graded as very achievable, achievable, possible, or low. These assessments were assigned by the author with the consensus of the AI experts interviewed.

2. Research Areas: Present cost estimating techniques will need modifications to predict AI systems. Each research area in Section 5 identifies a parameter that drives the cost to develop an AI system. Some parameters are already used in cost models and only need to be adjusted or extrapolated when estimating AI projects. Other parameters are not considered in present cost estimating models. In this case, data must be gathered, CERs analyzed, and models changed to accurately estimate AI projects.

   Each research area was evaluated for priority and doability and summarized in Table 6.1. Priority was assigned by how important the results of the research will be in providing competent and accurate cost estimates. Doability was assigned by how difficult it is to achieve research success. Again, these assessments were assigned by the author with the consensus of the AI experts interviewed.

Table 6.1
SUMMARY OF PRIORITY/DOABILITY

DISCUSSION AREA		PRIORITY	DOABILITY
T1	VHSIC	High	Possible
T2	Problem Solving Techniques	Extreme	Very Achievable
T3	Knowledge Representation	Extreme	Very Achievable
T4	AI Languages	Low	Possible
T5	Commercial Tools	Extreme	Very Achievable
T6	System Specifications	Extreme	Possible
T7	Development Methodology	Extreme	Very Achievable
T8	Schedule/Cost Estimating	High	Achievable
T9	Technology Roadmap	Extreme	Very Achievable
R1	Expanding Present Cost Models	Extreme	Achievable
R2	AI Cost Data Base	Extreme	Achievable
R3	Schedule Estimating	High	Possible
R4	Quality Estimating	High	Low
R5	Integration Cost	Low	Achievable
R6	Maintenance Cost	Low	Achievable
R7	Security Cost	Medium	Possible
R8	Computer Trade-off	Low	Very Achievable
R9	Level of Effort Cost	High	Achievable
R10	Knowledge Capture Cost	High	Achievable
R11	Memory Management Cost	Low	Low
R12	Interface Cost	Medium	Possible
R13	AI Language Proficiencies	High	Very Achievable
R14	Tools	High	Very Achievable
R15	Personnel Considerations	Medium	Achievable
R16	Program Management	High	Possible

INTERACTION AMONG AREAS

Technology areas identified and described in Section 4 relate to each other and also to research areas described in Section 5. This interaction is summarized in Table 6-2 and is helpful when planning the sequence of research projects proposed by this report. For instance, Technology area T3 describes AI Knowledge Representation which directly impacts the efficiency of the different AI Problem Solving Techniques discussed in technical area T2. Both research studies to understand these two areas may be performed at the same time or in a series. Research areas related to T2 and T3 are integration cost (R5), maintenance cost (R6) and level of effort predictions (R9). The conclusions from these research areas will compliment the conclusions from the technology research areas and so the research studies should be performed after the technology studies. Once the sequence of studies is determined, the point of entry must be identified to plan when each is to be started.

POINTS OF ENTRY

Technology impacts on AI systems development can occur two ways. Technology can evolve over time or it can breakthrough at a future point in time. In either case, the estimator must be able to understand the impact that the technology will have on system efficiency and cost. Table 6.3 summarizes the expected technology entry dates of the nine areas discussed in Section 4. The technology entry date is a point in time that technology will be understood well enough by cost estimators that a CER could be analyzed and calculated. That is not to say that analyzing and determining each CER will be easy. The ease to calculate any estimating relationships was listed in Table 6.1 as doability. Prior to any area's point of entry date, any tentative CER should be viewed as low in its reliability to predict quality cost estimates.

Research points of entry are also indicated on Table 6.3. This research entry date is a point in time when enough data has been analyzed by the cost estimator that new CERs can be accurately calculated or existing CERs can be competently modified. Gathering this data will occur over several years where the data base would evolve as additional data points become available. For instance, schedule estimating studies would be a long term effort because more AI systems need to be implemented and more scheduling data points analyzed before a reliable CER can be developed.

Data from research could also become available when system characteristics are quickly identified and CER's quickly calculated. For instance, when VHSIC technology is inserted in 1989, the existing CER interface costs could drastically changed.

These entry point dates were supplied by the author and agreed to by consensus of the experts interviewed. The purpose of these dates is to identify research areas that may be funded and tasked in future years when technology is hopefully better understood and enough data is available to predict cost relationships. Specific study recommendations and their estimated costs are presented next.

Table 6.2
SUMMARY OF RELATED AREAS

DISCUSSION AREA		RELATED AREA
T1	VHSIC	R1, R2, R12
T2	Problem Solving Techniques	T3, R5, R6, R9
T3	Knowledge Representation	T2, T6, R5, R6, R10
T4	AI Languages	R13
T5	Commercial Tools	T3, T4, R1-R16
T6	System Specifications	T7, R1, R2
T7	Development Methodology	T6, R1, R2, R3, R16
T8	Schedule/Cost Estimating	T9, R1, R2
T9	Technology Roadmap	R1-R16
R1	Expanding Present Cost Models	T1, T5, T6, T7, T8
R2	AI Cost Data Base	T1, T5, T6, T7, R1, R5, R6
R3	Schedule Estimating	T2, T7, T9
R4	Quality Estimating	T6, T7, R5, R6
R5	Integration Cost	T5, R2
R6	Maintenance Cost	R2
R7	Security Cost	R2
R8	Computer Trade-off	T2, T4, T5, R2, R11, R12, R13, R14
R9	Level of Effort Cost	R15, R16
R10	Knowledge Capture Cost	T2, T3, R1, R2, R3, R16
R11	Memory Management Cost	T1
R12	Interface Cost	T5, R2, R5, R6, R14, R16
R13	AI Language Proficiencies	T4, R14
R14	Tools	T2, T3, T5, R13
R15	Personnel Considerations	R3, R4, R6
R16	Program Management	T1-T9, R1-15

Table 6.3
AI/ES POINTS OF ENTRY

DISCUSSION AREA		1987	1988	1989	1990	1991	BEYOND
T1	VHSIC			X			
T2	Problem Solving Techniques	X					
T3	Knowledge Representation	X					
T4	AI Languages			X			
T5	Commercial Tools	X					
T6	System Specifications			X			
T7	Development Methodology	X					
T8	Schedule/Cost Estimating		X				
T9	Technology Roadmap	X					
R1	Expanding Present Cost Models		X				
R2	AI Cost Data Base		X				
R3	Schedule Estimating			X			
R4	Quality Estimating				X		
R5	Integration Cost		X				
R6	Maintenance Cost		X				
R7	Security Cost			X			
R8	Computer Trade-off	X					
R9	Level of Effort Cost		X				
R10	Knowledge Capture Cost		X				
R11	Memory Management Cost				X		
R12	Interface Cost			X			
R13	AI Language Proficiencies	X					
R14	Tools	X					
R15	Personnel Considerations		X				
R16	Program Management			X			

SPECIFIC RECOMMENDATIONS

We at Tecolote Research, Inc. contend that the best study approach to AI technology can best be summed up as "think big, start small." We recommend this as a practical approach that will minimize risk and disappointment when studying this new and rapidly evolving technology. The recommended studies listed in the following paragraphs were evaluated for a least risk, highest payoff, lowest cost in order to deliver practical, non-trivial cost data in a short amount of time. As technology evolves over the next years, these studies should be reviewed to enhance cost models as additional cost data is available and new AI cost parameters identified.

The priority rankings in Table 6.1 give the order that the recommended studies should be performed. Extreme priority calls for an immediate study. A high priority calls for a study late in 1987 or 1988 fiscal year. Medium and low priorities are areas that should be explored as funding permits in 1989 and beyond.

Table 6.4 presents each study area and the estimated level of effort in man months for a Senior Cost Analyst to accomplish it. Most of the studies will require tasking over several years because results will evolve as new technology is developed or additional data becomes available. AI technology is still in its infancy because there are only several hundred systems that have been developed. Many of those systems were implemented with R&D funds and will probably not have any cost accounting data. The systems developed with operational funds should have budget and cost data but there are only a few of these systems which means the number data sets will be small.

This is not to say, that the effort to develop AI cost estimating tools should be postponed until more data is available. Some information can be gathered now and initial software CERs determined. As shown in Table 6.4 there are eight extreme priority study areas that need to be pursued in 1987. They are as follows:

1. Problem Solving Techniques (T2). During 1987, it is recommended that a 3 man month study be performed in this area. Some data already exists about cost differences when various logic strategies are used to solve problems. Next year, this available data should to be analyzed and a primitive problem solving CER determined. A technology breakthrough is anticipated shortly. Industry is investing heavily in researching problem solving methods so that within two years, an AI engineer should be able to choose the best problem solving technique for any situation. Therefore, during 1988 and beyond, the crude problem solving CER will be enhanced and refined as more projects are developed and as new data is received.

2. Knowledge Representation (T3). During 1987, it is recommended that a 3 man month study be performed in this area to correlate knowledge representation and development cost and produce a primitive but important CER. Again, this is an area that will be well understood within several years because industry is investing in researching which representation technique best support which problem solving techniques. It is expected that this new research data will be

Table 6.4
RESEARCH/STUDY LOE ESTIMATES

(Man months of Senior Cost Research Labor)

DISCUSSION AREA		1987	1988	1989	1990	1991
T1	VHSIC		3	3	1	1
T2	Problem Solving Techniques	3	6	3	1	1
T3	Knowledge Representation	3	6	6	3	3
T4	AI Languages				3	2
T5	Commercial Tools	6	3	3	2	2
T6	System Specifications	6	6	3	3	3
T7	Development Methodology	6	6	3	3	3
T8	Schedule/Cost Estimating		3	3	3	3
T9	Technology Roadmap	3	3	3	2	1
R1	Expanding Present Cost Models	6	12	6	6	6
R2	AI Cost Data Base	24	24	12	12	12
R3	Schedule Estimating		3	3	2	2
R4	Quality Estimating		3	6	3	3
R5	Integration Cost				3	2
R6	Maintenance Cost			3	2	1
R7	Security Cost			3		
R8	Computer Trade-off				3	
R9	Level of Effort Cost		3	2		
R10	Knowledge Capture Cost		6	3		
R11	Memory Management Cost				1	
R12	Interface Cost			2		
R13	AI Language Proficiencies		3	3		
R14	Tools		3	2	1	1
R15	Personnel Considerations			2		
R16	Program Management		3	2		

available during 1988 and can than be evaluated to refine any knowledge representation CER developed by then.

3. Commercial Tools (T5). During 1987, it is recommended that a 6 man month study be performed in this area. Many AI development tools will be implemented within the next several years and their impact on system development productivity and cost is immense. A tool CER will be refined each year as new tools become available and more cost data is analyzed.

4. System Specifications (T6). During 1987, it is recommended that a 6 man month study be performed in this area. Inaccurate definition of system characteristics make the notion of creditable cost estimates and formal verification meaningless. This research area will provide a methodology to write rigid AI system specifications so the developer can provide a cost estimate, the programmer can build the system, and the tester can verify the results. This will be an on-going study for the following years. A method to specify AI projects will need to be defined, written, monitored, and then adjusted as more AI projects are developed. Specification techniques and management tools that produce programs and successful products will be understood, then modeled, then enhanced over the next five years.

5. Development Methodology (T7). During 1987, it is recommended that a 6 man month study be performed to subdivide AI product development into an organization of functions and phases. This development methodology will define the system structure and enable development costs to be accounted. Once a program development plan is available, cost tracking by function is possible and this historical cost data can then be used for predicting costs of analogous AI systems. This will be an on-going study. A development standard and methodology will be modified as more AI systems are built and the development methodology becomes better understood.

6. Technology Roadmap (R9). During 1987, it is recommended that a 3 man month study be performed to develop an AI Technology Roadmap. The information is needed so that the cost estimator can prepare for technology impacts. There should be two approaches: one defining when technologies are expected; and, the second to define an estimate of that technology impact on system performance and efficiency. Many new AI capabilities are anticipated within the next five years and each will impact development costs. This roadmap will be reviewed annually and consider future cost estimating requirements.

7. Expanding Present Cost Models (R1). During 1987, it is recommended that a 6 man month study be performed to expand present S/W cost models to include the capability to predict costs of AI systems. Most present cost models are based on source lines of code but that is not a valid basis for AI estimates. That is not to say that present cost models should be ignored when estimating AI projects. Instead, what is proposed and why this study is required, is to modify present cost models to include AI cost drivers as input parameters. This will be a

long term effort over several years to study, modify, and enhance cost models as more AI development data becomes available.

8. AI Cost Data Base (R2). During 1987, it is recommended that a 24 man month effort be tasked to develop an AI cost data base. It is vitally important for the cost estimator to have a large number of data points and their associated cost drivers to develop quality CERs. This data base allows the cost estimator to evaluate existing cost methods, how well they extrapolate for AI systems, and to evaluate technology impact on development costs. Maintaining, modifying, and enhancing this data base will be an on-going task for at least the next five years.

As shown in Table 6.4, there are nine high priority study areas that need to be pursued beginning in 1988. They are discussed as follows:

1. VHSIC (T1). VHSIC is a subject of interest to most people interviewed. Some studies are presently underway including the ESD sponsored VHSIC Technology Transition Plan. Continued study of the cost impact and enhanced system capabilities that result from VHSIC is important.

2. Schedule/Cost Estimating (T8). This capability will be an important first study area once the AI cost data base has been initiated. The ability to estimate AI project schedule and required manpower is the fundamental concern of all the study areas proposed. Probably, data will not be available for schedule/cost analysis until 1988, otherwise this study would be initiated as soon as possible.

3. Schedule Estimating (R3). New tools to assist the manager schedule AI projects are needed. To study this area requires the AI cost data base and a capability to accurately specify systems requirements.

4. Quality Estimating (R4). S/W quality and metrics is an area of interest to all managers and developers alike. Tools to verify AI projects and methods to validate capabilities and performance need to be identified and implemented. This study would modify MIL-STD-2167 and MIL-STD-2168 to define AI development and its V&V methodology.

5. Level of Effort Cost (R9). Distribution of labor over the different development phases will be the basis of predicting project schedules This is a high interest study with a good pay-off.

6. Knowledge Capture Cost (R10). AI system success depends on the knowledge engineer gathering, understanding, and organizing the expertise of the expert. These development activities represent a major percentage of total system cost and yet data is not available to perform cost analysis. This study area needs to survey the techniques, costs, and results to gather and implement an AI knowledge base.

7. AI Language Proficiencies (R13). Which of the AI languages is the best to solve a particular application is a hotly debated topic

Unfortunately, most arguments are based on emotion and hearsay without much scientific foundation. This study will provide the data to evaluate languages, their features, and their ease to build different AI systems. This should be accomplished as two tasks a year apart in

8. Tools (R14). Many tools are expected to be introduced over the next several years that will allow AI systems to be developed quicker and easier. The cost impacts of each tool could be significant and must be anticipated and understood by the cost estimator.

9. Program Management (T16). This study will identify new tools, practices, and policies that management will require to perform cost trade-off studies, make/buy decisions, staff requirement plans, schedule plans, and corporate strategy plans. AI is a new technology that requires new approaches to some very traditional problems.

The remaining study areas identified in Table 6.4 have a medium or low priority. These areas are not being funded very heavily by industry as research projects and so their point of entry date is questionable. Additionally, AI state of the art is evolving rapidly which makes it difficult to accurately predict what will be the most important study areas in several years. For these two reasons, the medium and low priority study areas should be scrutinized again in two years to evaluate the necessity to fund those tasks and to identify any new studies depending on then existing AI state of the art.

APPENDIX A

BIBLIOGRAPHY

"Artificial Intelligence in 2000 A.D." Defense Electronics, July 1985, By Jan David Wald.

Boden, Margaret A., Artificial Intelligence and Natural Man, Basic Books, Inc., Publishers, 1977.

Boehm, Barry W., Software Engineering Economics, Prentice-Hall Inc., 1981.

Brooks, Frederick P., Jr., The Mythical Man-Month, Addison-Wesley Publishing Company, 1982.

"Development of Software for Ballistic-Missile Defense", Scientific American, December 1985, by Herbert Lin.

"Expert Systems Promise Supersmart PCs", Popular Science, March 1986, by William J. Hawkins.

Feigenbaum, E., and Paul R. Cohen, The Handbook of Artificial Intelligence, 3 Vols., William Kaufman, Inc., 1982.

"Future Military Applications for Knowledge Engineering", RAND Note, N-2102-1-AF, by Steven C. Bankes, July 1985.

Harmon, Paul and David King, Expert Systems, Artificial Intelligence in Business, John Wiley & Sons, Inc., 1985.

IEEE 1985 National Aerospace and Electronics Conference, May 1985, 2 Vols.

Nagy, Tom and Dick Gault and Monica Nagy, Building Your First Expert System, Ashton-Tate, 1985.

Nilsson, Nils J., Problem Solving Methods in Artificial Intelligence, McGraw-Hill Book Company, 1971.

"Power Tools for Programmers", Datamation, February 1983, by Beau Sheil.

Report to the Congress on the Strategic Defense Initiative, 1985.

Report to the Director Strategic Defense Initiative Organization, Summer Study 1985, Eastport Study Group.

Report of the Study on Eliminating the Threat Posed by Nuclear Ballistic Missles, February 1984, James E. Fletcher, Study Chairman.

Rich, Elain, Artificial Intelligence, McGraw-Hill Book Company, 1983.

"Software Aspects of Strategic Defense Systems", Communications of the ACM, December 1985, by David Lorge Parnas.

"Software Tools Speed Expert Systems Development", High Technology, March 1985, by Paul Kinnucan.

Weizenbaum, Joseph, Computer Power and Human Reason, W.H. Freeman and Company, 1976.

"Why Expert Systems Fail", Journal of the Operational Research Society, Vol. 36 No. 7, by Michael Z. Bell.

Winston Patrick Henry and Karen A. Prendergast, AI Business, MIT Press, 1985.

Winston, Patrick Henry, Artificial Intelligence, Addison-Wesley Publishing Company, 1984.

Winston, Patrick Henry and Berthold Klaus Paul Horn, LISP, Addision Wesley Publishing Company, 1984.

APPENDIX B

LIST OF AI EXPERTS

JPL

Yukio Nakamura	(818)354-7344
John Tupman	577-9270
Dr. Steven Vere	577-9280
John Battenburg	354-6852
Carl Solloway	577-9412
Ken Holmes	577-2879
David Clough	577-6480
Wun Chiou	577-8444
Gene Cynkin	
Bert Hansen	
Joe Hanson	
Fredrick Thompson	
Robert Mackin	577-8499
Paul Schenker	
Sven Grenader	

MITRE - Bedford, MA

Myra Prelle	(617)277-7752

Institute of Defense Analysis - Arlington, VA

Ann Douville	(703)845-3508
Mike Bloom	(703)824-5512
Dr. John Salasin	
John Quadrata	(703)845-3508

WPAFB - Dayton, OH

Lt. Jeff Commer	(513)255-7388
Lt. John Jones	(513)255-6347

USAF SD - Los Angeles, CA

Capt. Paul Killingsworth	(213)643-1772

Information Sciences Institute - Marina del Rey, CA

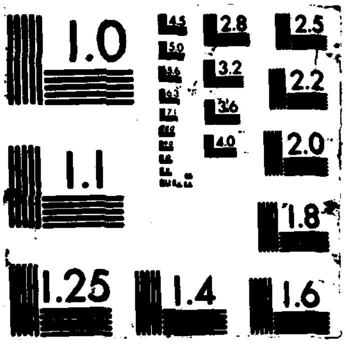
Leonard Friedman	(213)822-1511

TRW - Redondo Beach, CA

Barry W. Boehm	

DEC - Maynard, MA	
Steve Gutz	(617)568-6966
Suzanne McGuire	(617)568-6944
MITRE - McClean, VA	(703)883-6000
Russ Bonasso	
Jim Antonisse	
Robert Fleming	
Dr. G. Emami	
Dr. Sharon Laskowski	
Richard Nugent	
Harold Ottoson	
DARPA - Rosslyn, VA	
Albert E. Brandenstein	(202)694-1702
Barbara Weather	(202)692-6083
United Technologies Norden Systems	
Steve Hall	(213)615-0241
Rich Crane	(213)615-0241
Lee Best	(714)250-3311
Ed Miller	(800)243-5840
RADC - Griffis AFB, NY	
Andrew Chrusciciki	(315)330-4063
Dr. Fred I. Diamond	(315)330-4512
Mr. R. Urts	(315)330-2167
Don Roberts	(315)330-2973
Dr. Northrop (Nort)Fowler	(315)330-7794
Ziggy Pankowicz	(315)330-3241
Texas Instruments - Los Angeles, CA	
Harold Stover	(213)217-7055
Logicon - Los Angeles, CA	
Eric Carson	(213)831-0611
Rand Corp. - Los Angeles, CA	
Dr. Phillip Klohr	
Steven C. Bankes	

AD-A186 339 ARTIFICIAL INTELLIGENCE/EXPERT SYSTEM COST RESEARCH 2/2
ROADMAP(U) TECOLOTE RESEARCH INC SANTA BARBARA CA
S STEPNEK SEP 86 CR-0145 ESD-TR-87-168 F19628-84-D-0019
UNCLASSIFIED F/G 12/9 NL


1.0
4.5
5.0
5.6
6.3
7.1
8.0
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

HAC - Los Angeles, CA

Doug Partridge
Bill Dahl
Harriet Kagiwada

RDA Assoc. - Marina del Rey, CA

Genny Burke	(213)822-1715
Keith Uncapher	

Aerospace Corp. - Los Angeles, CA

Dr. Michael Mellea	
Warren Lamb	
Peter Covan	
Peter Coffee	(213)647-1500

Software Architecture & Engineer, Inc. - Arlington, VA

John Sapp	(703)276-7910
Geoffrey Orr	(703)284-3811
Tony Magliero	(703)284-3812

Kestrel Research Institute - Palo Alto, CA

Cordell Greens

Reasoning Systems - Palo Alto, CA

John Antoine

MIT

Marvin Minsky

BB&N

Daniel Bobrow

SDC

Dr. David Serls	(215)648-7456

Intellicorp

Jeffry Magill	(617)577-1067

Kurzviel

Phil Fitzpatrick	(301)982-0666

Sperry

Fred Hills	(703)556-5762
John Esch	(703)556-5762

Intelligenceware, Inc. - Los Angeles, CA

Dr. Kamran Parsaye

APPENDIX C

AI VENDORS

SEAI Technical Publications
P.O. Box 590
Madison, GA 30650
(401)342-9638

Carnegie Group
650 Commerce Court
Station Square
Pittsburg, PA 15219
(412)642-6900

Expert Systems International
1700 Walnut Street
Philadelphia, PA 19103
(215)735-8510

Kemp-Carraway Heart Institute
1600 North 26 Street
Birmingham, AL 35234
(205)226-6697

Level Five Research, Inc.
503 Fifth Avenue
Indialantic, FL 32903
(305)729-9046

Logicware, Inc.
1000 Finch Avenue West, Suite 600
Toronto, Ontario
(416)665-0022

Applied Expert Systems
Five Cambridge Center
Cambridge, MA 02142
(617)492-7322

Venrock Associate
30 Rockefeller Plaza, Suite 5508
New York, NY 10012
(212)247-3700

Texas Instruments, Inc.
12501 Research Blvd., M/S 2244
Austin, TX 78759

Software A&E
1600 Wilson Blvd.
Arlington, VA 22209
(703)276-7910

Ion Technology Services, Inc.
426 E. Boston Street
Topsfield, MA 01983
(800)221-2394

Artelligence, Inc.
14902 Preston Road, Suite 212-252
Dallas, TX 75240
(214)437-0361

Programs In Motion Inc.
10 Sycamore Road
Wayland, MA 01778
(617)653-5093

Inference Corporation
5300 W. Century Blvd.
Los Angeles, CA 90045
(213)417-7997

Expertelligence
559 San Ysidro Road
Santa Barbara, CA 93108
(805)969-7874

AIRUS
11830 SW Kerr Parkway
Lake Oswego, OR 97034
(503)246-1782

XMCO Inc.
8200 Greensboro, Suite 801
McLean, VA 22102
(703)648-9600

SRI
333 Ravenswood Avenue
Menlo Park, CA 94025
(415)326-6200

General Research Corporation
5383 Hollister Avenue
Santa Barbara, CA 93111
(805)964-7724

Symbolics, Inc.
9600 DeSoto Avenue
P.O. Box 705
Chatsworth, CA 91311
(818)998-3600

Tiburon Systems
611 River Oaks Parkway
San Jose, CA 95134
(408)371-9400

PAR Technical Corporation
7926 Jones Branch Drive
McLean, VA 22102
(703)478-9690

TYMSHARE
20705 Valley Green Drive
Cupertino, CA 95014
(408)446-6000

Xerox Artificial Intelligence Systems
250 N. Halstead Street
P.O. Box 7018
Pasadena, CA 91109
(818)351-2351

Advanced Information & Decisions Sys.
201 San Antonio Circle, Suite 286
Mountain View, CA 94040
(415)965-1008

APPENDIX D

TRIP/CALL REPORTS

This appendix provides the notes from telephone and visit interviews with AI managers, developers, and government personnel. Reports are presented in alphabetical order as follows:

Battenburg, John A.	JPL
Bloom, Mike	Institute for Defense Analysis
Boehme, Barry	TRW
Brandenstein, Al	Darpa
Chrusciciki, Andy	RADC, CDEE Branch
Coffee, Peter	Aerospace Corporation
Davis, Jim	Arthur D. Little, Inc.
Douville, Ann	Institute for Defense Analysis
Friedman, Leonard	Information Scientist Institute
Gutman, Abraham	Inference Corporation
Killingsworth, Capt. Paul	USAF Space Division
Killingsworth, Capt. Paul	USAF Space Division
Magliero, Tony	Software A & E
Nakamura, Yukio	JPL
Orr, Geoffrey	Software A & E
Pankowicz, Ziggy	RADC, COTC Branch
Piketty, Laurent	Inference Corporation
Roberts, Don	RADC, COE Branch
Sapp, John	Software A & E
Slavinski, Dick	RADC, COE Branch
Talada, Maj. Don	RADC, COT Branch
Tupman, John	JPL
Urts, Ray	RADC, CO Division
Vera, Dr. Steven A.	JPL